SYSTEMUNTERLAGEN-DOKUMENTATION	ANLEITUNG FUER DEN PROGRAMMIERER	MOS
Stand: 01.05.1982	EIEX 1521	K 1520

ANLEITUNG FUER DEN PROGRAMMIERER

Internspeicherorientiertes
Echtzeitsteuerprogrammsystem
EIEX 1521

VEB Robotron
Zentrum fuer Forschung und Technik

Dok.-Nr. C0264-0059-1 M 1130

SYSTEMUNTERLAGEN- DOKUMENTATION	ANLEITUNG FUER DEN PROGRAMMIERER	MOS
Stand: 01.05.1982	EIEX 1521	K 1520

ANLEITUNG FUER DEN PROGRAMMIERER

Internspeicherorientiertes
Echtzeitsteuerprogrammsystem
EIEX 1521

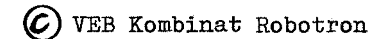
VEB Robotron
Zentrum fuer Forschung und Technik

Dok.-Nr. C0264-0059-1 M 1130

Die vorliegende Systemunterlagen-Dokumentation entspricht dem Stand vom 01.05.1982.

Die Ausarbeitung der Unterlagen erfolgte durch ein Kollektiv des VEB Robotron Zentrum fuer Forschung und Technik.

Im Interesse einer staendigen Weiterentwicklung der Systemunterlagen werden alle Leser gebeten, ihre Vorschlaege bzw. Hinweise zur Verbesserung dem Herausgeber mitzuteilen.



Vorwort

Das internspeicherorientierte Echtzeitsteuerprogrammsystem EIEX 1521 erlaubt eine einfache Kommunikation von Programmierer und Bediener mit dem MRS K 1520 bei der Realisierung von anwendungsspezifischen Programmsystemen im Echtzeitbetrieb.
Es schafft die notwendigen Voraussetzungen, sodass der Anwender ohne spezielle Kenntnis der Geraetetechnik das Mikrorechnersystem voll nutzen und sich auf die Entwicklung seiner anwenderspezifischen Programme konzentrieren kann.
Voraussetzung fuer das Verstaendnis der Schrift ist die Kenntnis der:

- Betriebsdokumentation Mikrorechner K 1520

- Sprachbeschreibung Assemblersprache SYPS K 1520
  Dok-Nr. 1.78.519.030.0/78

- Beschreibung des Systems ursadat 5000

- Anwendungsbeschreibung Internspeicherorientiertes Echtzeitsteuerprogrammsystem EIEX 1521
  Dok-Nr. C0262-0058-1 M1130

Inhaltsverzeichnis Seite

Anlagen

Verzeichnisse

1. Allgemeine Hinweise

1.1. Umfang und Aufbau der Systembeschreibung

Die Beschreibung des Echtzeitsteuerprogrammsystems EIEX 1521 beinhaltet eine detaillierte Darstellung seiner Systemkomponenten und deren wesentliche Funktionen. Diese dient dem Problemanalytiker und Programmierer bei der Konzipierung und Programmierung von Applikationsprogrammsystemen als Arbeitsmittel.
Die einzelnen Abschnitte der Beschreibung sind einheitlich so gegliedert, dass zunaechst ein Ueberblick zur Wirkungsweise der Systemkomponente gegeben wird und nachfolgend die fuer die Programmierung wesentlichen Einzelheiten dargestellt werden. Jeder Abschnitt wird mit der Ruf- und Kommandobeschreibung abgeschlossen. Angaben ueber Fehlererkennung und -behebung werden nicht gesondert aufgefuehrt.
Bei einzelnen Abschnitten entfallen diese Angaben, da keine EIEX-Rufe oder Kommandos zu diesen Komponenten existieren.

1.2. Benutzung der EIEX-Rufe

Durch Einfuegen von Steuerprogrammrufen in ein Applikationsprogramm wird veranlasst, dass dieses mit dem EIEX 1521 in Verbindung tritt, um Systemfunktionen auszufuehren.
EIEX 1521 bearbeitet max. 128 Rufe, die vom Anwender aus den standardmaessig bereitgestellten Ruf-Routinen ausgewaehlt, aber auch durch anwendereigene Ruf-Routinen unter Beachtung der Anschlussbedingungen ergaenzt werden koennen.
Die allgemeine Form eines EIEX-Rufes in der Backus-Notation lautet:

Namensfeld	Operationsfeld	Parameterfeld	Kommentarfeld
[name]	ruf	(par...!paradr)	[kommentar]

Erlaeuterung:

Fuer die Kodierung der EIEX-Rufe gelten die Regeln der Assemblersprache SYPS K 1520.

name – 1 bis 5 Buchstaben oder Ziffern
Mindestens das 1. Zeichen muss ein Buchstabe sein.
Als Abschluss muss in jedem Fall ein Doppelpunkt stehen.

ruf – Angabe des EIEX-Rufes

par – Angabe von Parametern entsprechend dem zu kodierenden EIEX-Ruf

paradr – Angabe der Adresse, unter der die Parameter im Block notiert sind, in symbolischer oder direkter Form

kommentar – Angabe eines Kommentars
Er beginnt im Normalfall nach dem Parameterfeld durch Kennzeichnung mit einem Semikolon.
Eine ganze Programmzeile gilt als Kommentarzeile, wenn als 1. Zeichen ein Semikolon steht. Anschliessend ist eine Eingabe von maximal 70 Zeichen moeglich.Werden in einer Kommentarzeile vor dem Semikolon mindestens 1 Leerzeichen oder Tabulator erfasst, so wird diese Zeile als eingerueckte Kommentarzeile gewertet. Die Kommentarlaenge betraegt dann max. 60 Zeichen.

Ein wesentlicher Parameter eines jeden Rufes ist die Registerrettungsart.
Mit deren Angabe werden vor Abarbeitung des Rufes die gewuenschten Register des Applikationsprogramms automatisch gerettet und nach Ausfuehrung der Rufroutine regeneriert.
Folgende Arten der Registerrettung im Task-Stack werden unterstuetzt:

R = 0 – keine Registerrettung
R = 1 – Rettung des Hauptregistersatzes und der Indexregister

R = 2 - Rettung des Haupt- und Tauschregistersatzes sowie der Indexregister

Ein Ruf wird allgemein im Maschinencode des Applikationsprogrammes wie folgt abgespeichert:

RST	n	n = 8,16,24
Rufnummer	m	m = 0...127
Parameterfolge oder Parameteradresse		

Bei Angabe einer Parameteradresse im Ruf betraegt die Ruflaenge im Maschinenprogramm konstant 4 Byte, sonst ist sie variabel und von der Anzahl der folgenden Parameter abhaengig. Bei einer Rufnotierung mit Angabe der Parameteradresse wird das Bit 7 im Rufnummer-Byte gesetzt.Diese Form der Rufnotation ist nur sinnvoll, wenn die Ruflaenge 4 Byte ueberschreitet oder im AP die gleichen Rufparameter wiederholt durch Rufe benutzt werden.
Fuer die Benutzung von EIEX-Rufen muss folgende allgemeingueltige Vorschrift beachtet werden:

- In Applikationsprogrammen und AP- spezifischen Unterprogrammen duerfen alle EIEX-Rufe benutzt werden.
- In Bibliotheksunterprogrammen duerfen keine EIEX-Rufe benutzt werden.
- In Interrupt-Service-Routinen duerfen nur die EIEX-Rufe RUN, GO, DISP, ENAP, CNCL, DATE und TIME benutzt werden.

Eine Uebersicht der in EIEX 1521 enthaltenen Rufe ist in den Anlagen 1 (Uebersicht der EIEX-Rufe) und 4 (Schluesselwortparameter fuer Rufe und Kommandos) enthalten.

1.3. Benutzung der EIEX-Kommandos

Der Bediener eines mit EIEX 1521 arbeitenden Betriebssystems kann durch Eingabe von Kommandos mit diesem in Verbindung treten. Jedes Kommando besteht aus der Kommandobezeichnung und den

dazugehoerigen Parametern. Vom Anwender koennen aus den standardmaessig bereitgestellten Kommandos die fuer seinen Anwendungsfall relevanten Kommandos ausgewaehlt werden. Weiterhin ist eine Ergaenzung durch anwendereigene Kommandos unter Beachtung der Anschlussbedingungen moeglich. Die allgemeine Notation eines EIEX-Kommandos lautet:

Kommandobezeichnung	Parameter
kom	par...

Erlaeuterung:
kom - Angabe des EIEX-Kommandos
par - Angabe von Parametern entsprechend dem EIEX-Kommando

Die eingetasteten Kommandos werden syntaktisch geprueft und in einen Ruf transformiert, der in einem Applikationsprogramm (Kommando-Task) hoher Prioritaet einmalig zur Abarbeitung gebracht wird. Die Kommandos sind aufgabengebunden den entsprechenden Systemkomponenten zugeordnet. Die prinzipielle Arbeitsweise der Kommandobearbeitung wird unter Pkt.3.7, Organisation der Bedienerkommunikation, erlaeutert.
Die geraetetechnische Voraussetzung fuer die Eingabe von Kommandos bildet die interruptgesteuerte alphanumerische Tastatur K 7602. Als Ausgabemedium ist der Bildschirm MON 1 K 7221 vorgesehen.
Eine Uebersicht zu den in EIEX 1521 enthaltenen Kommandos ist in der Anlage 2 (Uebersicht der EIEX-Kommandos) enthalten.

1.4. Syntaktische Regeln

Bei der Beschreibung der EIEX-Rufe und Kommandos werden folgende syntaktische Regeln verwendet:
- Grossgeschriebene Zeichen bzw. Worte sind Grundsymbole der

EIEX-Rufe und Kommandos.
Diese Zeichen definieren sich selbst, sie werden in der angegebenen Form codiert.

- Kleingeschriebene Zeichen bzw. Worte sind durch Werte zu ersetzen, die der Beschreibung der EIEX-Rufe und Kommandos zu entnehmen sind.
- Ausrufezeichen !
  Das Ausrufezeichen steht zwischen alternativen Elementen, von denen genau eines ausgewaehlt werden muss.
- Runde Klammern ( )
  Runde Klammern enthalten eine horizontale Aufzaehlung von Moeglichkeiten, die durch das Ausrufezeichen getrennt sind.
- Eckige Klammern [ ]
  Eckige Klammern enthalten wahlweise angebbare Parameter

2.Geraetetechnische Festlegungen

Das Mikrorechnersystem K 1520 stellt ein Baugruppensystem dar. Aus einem oder mehreren Zentrale - Recheneinheit - Moduln, Speichermoduln und Anschluss - Steuerungen sowie Zusatzbaugruppen , die ueber einen gemeinsamen Rechnerbus zusammenarbeiten, koennen Geraetekonfigurationen entsprechend dem Einsatzfall zusammengestellt werden .
Das internspeicherorientierte Echtzeitsteuerprogramm EIEX 1521 unterstuetzt den Einsatz von den nachfolgend aufgefuehrten OEM-Baugruppen.

2.1. Zentrale Recheneinheit

ZRE - Typen : K 2521
Speicher - Typen: K 3520, K 3521, K 3525, K 3620, K3621, K 3820

Die Auswahl der Speichersteckeinheiten ist abhaengig vom Einsatzfall des Anwenders.

2.2. Periphere Geraete

2.2.1. Datenverarbeitungsperipherie

Durch EIEX 1521 werden auf der Basis der Anschluss - Steuerung K 6022 die peripheren Geraete

- Lochbandleser 1210-0333
- Lochbandstanzer 1215-1011
- Seriendrucker 1156

ueber das SIF- 1000 Interface bedient. Weiterhin werden die

- Bildschirmanzeigebaugruppe MON1 K 7221
  (Anschlusseinheit ABS K 7023)
- Folienspeicher MF 3200
  (Anschlusseinheit AFS K 5121)
- alphanumerische Tastatur K 7602
  (Anschlusseinheit ATD K 7026)

durch EIEX 1521 unterstuetzt.

2.2.2. Prozessperipherie

Durch EIEX 1521 werden die Prozess - Steckkarten des Systems ursadat 5000 programmtechnisch unterstuetzt.
Sie beinhalten:

- Digitale Ein- und Ausgaben statisch und dynamisch
- Analoge Ein- und Ausgaben

- Impulsein- und ausgaben
- Serielles Zwischenblock-Interface

Eine Uebersicht zu den durch EIEX 1521 unterstuetzten Prozesssteckkarten aus dem System ursadat 5000 wird in Anlage 5 (Prozess- Steckkarten ursadat 5000) gegeben.

3. Arbeit mit dem Echtzeitsteuerprogrammsystem EIEX 1521

3.1. Vorrangorganisation der Applikationsprogramme

3.1.1. Vorrangsystem der Applikationsprogramme

Alle vom Anwender selbst entwickelten oder mittels problemorientierter Systemunterlagen erarbeiteten Programme werden als Applikationsprogramme bezeichnet. Die im folgenden beschriebene Organisationsform gilt jedoch nicht nur fuer diese Programme, sondern auch fuer einige Programme des EIEX 1521 selbst (z.B. Ausgabe von Systemnachrichten).
Jedes Applikationsprogramm wird als Arbeitsaufgabe (Task) von dem Steuerprogramm aufgefasst, die in das Vorrangsystem des EIEX 1521 durch die Angabe seiner Prioritaet (Tasknummer) eingeordnet ist und damit eine feste Position bei der Taskverwaltung erhaelt. Die Applikationsprogramme werden nicht ueber einen symbolischen Namen, sondern ueber die Tasknummer, die identisch mit der festgelegten Prioritaet des Applikationsprogramms im AP- System ist, durch die Vorrangorganisation gesteuert und verwaltet.
Jede Task ist somit durch die Angabe einer Nummer eindeutig festgelegt. Die Prioritaet des Applikationsprogrammes steigt mit fallender Tasknummer, die bei EIEX-Rufen bzw. Kommandos stets positiv ganzzahlig anzugeben ist.

Durch das Vorrangsystem koennen maximal 255 Task verwaltet werden, von denen zwei Task durch EIEX 1521 fuer die Kommando- und Systemnachrichtenorganisation belegt sind und somit dem Anwender nicht zur Verfuegung stehen.
Das interne Abbild der Vorrangstruktur des EIEX 1521 sind die Taskanmelde- und die Tasksteuerregister. Sie geben Auskunft ueber den derzeitigen Zustand jeder Task (zum Beispiel ausser Betrieb, aktiviert, Wartezustand usw.). Das Vorrangsystem wird durch die Entschluesselung der Taskanmelde- und Tasksteuerregister verwirklicht.
Die Entschluesselung beginnt bei der Task mit der hoechsten Prioritaet im System (Task 1). Wird eine Task gefunden, die angemeldet ist, wird deren Bearbeitung eingeleitet.
Wenn keine Task angemeldet ist, steht EIEX 1521 solange im dynamischen Stop, bis ein Interrupt (z.B. durch die Uhr) zu einer Taskanmeldung fuehrt. Dieser absolute Vorrang gilt fuer das gesamte Echtzeitsteuerprogramm, da nur die Anmeldung einer Task mit hoher Prioritaet (niedrige Tasknummer) zur Unterbrechung einer Task mit niedriger Prioritaet (hohe Tasknummer) fuehrt.

3.1.2. Hinweise zum Aufbau des Applikationssystems

Das Einordnen der Programme in die Vorrangorganisation erfolgt beim Binden durch die Angabe der Tasknummer.
Durch EIEX 1521 wird vorausgesetzt, dass jede Task einen eigenen Stack besitzt und dessen Anfangsadresse sowie die Taskstartadresse beim Binden in die Vorrangorganisation eingeordnet wird.
Der bestimmende Gesichtspunkt fuer die Einordnung in die Vorrangstruktur ist die erforderliche Reaktionszeit der einzelnen Tasks. Zweckmaessig sollten Tasks mit einer geringen Reaktionszeit eine hoehere Prioritaet gegenueber denen mit einer grossen Reaktionszeit haben. Als Reaktionszeit wird die Zeit verstanden, die von der Anmeldung der Task bis zu deren Bearbeitungsende vergehen darf.

weitere Kriterien fuer die Einordnung der Task sind deren Bearbeitungszeit und Zykluszeit. Als Bearbeitungszeit gilt dabei nur die Zeit, in der die Task ununterbrochen bearbeitet wird. EIEX-Rufe, die zu einem Wartezustand der Task fuehren, beenden die ununterbrochene Bearbeitung. Allgemein gilt:

hohe Bearbeitungszeit	–	niedrige Prioritaet
geringe Bearbeitungszeit	–	hohe Prioritaet
grosse Zykluszeit	–	niedrige Prioritaet
kleine Zykluszeit	–	hohe Prioritaet

Fuer die Einordnung einer zyklisch zu aktivierenden Task innerhalb eines AP-Systems gilt, dass die Summe der Bearbeitungszeiten aller Tasks mit hoeherer Prioritaet kleiner als die Zykluszeit der Task sein muss.

3.1.3. EIEX-Rufe und Kommandos

3.1.3.1. Starten einer Task RUN

a) Aufgabe: Sofortiger einmaliger Start einer Task
b) Status: Ruf oder Kommando
c) Schreibweise:
Kommando:
RUN TASK = p
Ruf:
[name] RUN R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TASK	– Applikationsprogramm	
p	– Prioritaet der Task	$0 < p < 256$
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH

symb - symbolische Adresse

d) Ruf-Nr.: 0
e) Ruflaenge: 4 Byte
f) Wirkung: Die Task wird sofort zur Bearbeitung angemeldet und entsprechend ihrer Prioritaet durch die Vorrangorganisation gestartet. Ist die zu startende Task nicht generiert, so erfolgt die Ausgabe einer Fehlerinformation auf dem Protokollgeraet und die den Ruf ausloesende Task wird fortgesetzt. Die gleiche Reaktion erfolgt ebenfalls, wenn die im Ruf enthaltene Task zu diesem Zeitpunkt verhindert ist.

3.1.3.2. Beenden einer Task BYE

a) Aufgabe: Abmelden einer beendeten Task bei der Vorrangsteuerung
b) Status: Ruf
c) Schreibweise:
Ruf:
[name] BYE R=n[,REP!PARA=(adr!symb)][;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
REP	- Wiederstarterlaubnis	
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr.: 1
e) Ruflaenge: 3 Byte

f) Wirkung: Dieser Ruf bildet den logischen Abschluss einer Task und beendet deren Abarbeitung. Bei Angabe des Parameters REP im Ruf erfolgt eine Pruefung, ob waehrend der Abarbeitung der Task eine erneute Anmeldung aufgetreten ist, die dann zu einem wiederholten Starten der Task fuehrt. Es wird dabei nur eine einmalige Anmeldung beruecksichtigt.

3.1.3.3. Fortsetzen einer Task GO

a) Aufgabe: Fortsetzen einer mit EIEX-Ruf PAUS unterbrochenen Task

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

GO TASK = p

Ruf:

[name] GO R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TASK	– Applikationsprogramm	
p	– Prioritaet der Task	$0 < p < 256$
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr.: 4

e) Ruflaenge: 4 Byte

f) Wirkung: Der Pausezustand fuer die im Ruf angegebene Task wird aufgehoben und die Fortsetzung ihrer Bearbeitung im Vorrangsystem angemeldet.

ist die im Ruf angegebene Task nicht generiert, erfolgt die gleiche Reaktion wie beim EIEX-Ruf RUN (siehe Pkt. 3.1.3.1, Starten einer Task).
Die Ausgabe einer Fehlerinformation erfolgt ebenfalls, wenn die im Ruf enthaltene Task sich nicht im Pausezustand befand bzw. deren Startzeitpunkt durch die Zeitorganisation verwaltet wird.

3.1.3.4. Verhindern der Taskbearbeitung DISP

a) Aufgabe: Verhindern der Bearbeitung einer Task

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

```
DISP  TASK = p
```

Ruf:

```
[name] DISP  R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \le n \le 2$
TASK	– Applikationsprogramm	
p	– Prioritaet der Task	$0 < p < 256$
PARA	– Parameteradresse	
adr	– absolute Adresse	$0 < adr <$ FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr.: 5

e) Ruflaenge: 4 Byte

f) Wirkung: Die im Ruf angegebene Task wird von der Bearbeitung bis auf Widerruf durch den EIEX-Ruf ENAP ausgeschlossen. Ist die Task zur Zeit in Arbeit, so wird sie abgebrochen. Eine bereits eingeleitete E/A-Operation fuer diese Task wird zu Ende

gefuehrt, weitere in den Warteschlangen eingetragenen E/A-Rufe fuer diese Task werden nicht mehr ausgefuehrt.
Ist die im Ruf angegebene Task nicht generiert, erfolgt die gleiche Reaktion wie beim EIEX-Ruf RUN (siehe Pkt. 3.1.3.1, Starten einer Task).

3.1.3.5. Erlauben der Taskbearbeitung ENAP

a) Aufgabe: Freigabe einer durch den EIEX-Ruf DISP verhinderten Task

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

ENAP TASK = p

Ruf:

[name] ENAP R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
TASK	- Applikationsprogramm	
p	- Prioritaet der Task	$0 < p < 256$
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr.: 6

e) Ruflaenge: 4 Byte

f) Wirkung: Fuer die im Ruf angegebene Task wird die Sperrung der Bearbeitungsmoeglichkeit aufgehoben.
Ihre Bearbeitung erfolgt bei einer erneuten Anmeldung ab der Startadresse der Task.

Ist die im Ruf angegebene Task nicht generiert, erfolgt die gleiche Reaktion wie beim EIEX-Ruf RUN (siehe Pkt. 3.1.3.1, Starten einer Task).

3.1.3.6. Austragen einer Task aus der Bearbeitung von der Vorrang- und Zeitorganisation CNCL

a) Aufgabe: Suspendieren der Bearbeitung einer Task im Steuerprogrammsystem durch Austragen aus der Vorrang- und Zeitorganisation

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

CNCL TASK = p

Ruf:

[name] CNCL R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
TASK	- Applikationsprogramm	
p	- Prioritaet der Task	$0 < p < 256$
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr.: 2

e) Ruflaenge: 4 Byte

f) Wirkung: Die im Ruf angegebene Task wird aus der Vorrang- und Zeitorganisation ausgetragen und abgebrochen. Eine bereits eingeleitete E/A-Operation wird zu Ende gefuehrt, weitere in den Warteschlangen eingetragene E/A-Rufe dieser Task werden nicht mehr ausgefuehrt. Ein erneuter Start der Task ist nur ueber den EIEX-Ruf RUN moeglich.

Ist die im Ruf angegebene Task nicht generiert, erfolgt die gleiche Reaktion wie beim EIEX-Ruf RUN (siehe Pkt. 3.1.3.1, Starten einer Task).

3.1.3.7. Unterbrechung der Abarbeitung einer Task PAUS

a) Aufgabe: Unterbrechen der Abarbeitung einer laufenden Task
b) Status: Ruf
c) Schreibweise:
Ruf:
[name] PAUS R=n[,(PRT!PARA=(adr!symb))][;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:
R - Registerrettung
n - Registerrettungsart $0 \leq n \leq 2$
PRT - Protokollierung des Pausezustandes auf dem Protokollgeraet
PARA - Parameteradresse
adr - absolute Adresse 0 < adr < FFFFH
symb - symbolische Adresse

d) Ruf-Nr.: 3
e) Ruflaenge: 3 Byte
f) Wirkung: Die Abarbeitung der laufenden Task wird unterbrochen, wobei der Pausezustand der Task auf dem Kommunikationsgeraet protokolliert werden kann. Die weitere Bearbeitung der Task erfolgt nur ueber den EIEX-Ruf GO.
Eine bereits eingeleitete E/A-Operation wird zu Ende gefuehrt, weitere in den Warteschlangen eingetragene E/A-Rufe dieser Task werden nicht mehr ausgefuehrt.

3.1.3.8. Wechsel der Prioritaet einer Task HELP

a) Aufgabe: Einmaliger Prioritaetswechsel einer Task mit einer fest vorgegebenen Prioritaet

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

HELP TASK = p

Ruf:

[name] HELP R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TASK	– Applikationsprogramm	
p	– Prioritaet der Task	$0 < p < 256$
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr.: 16

e) Ruf-Laenge: 4 Byte

f) Wirkung: Fuer die im Ruf angegebene Task wird ein Prioritaetswechsel auf eine festgelegte Prioritaet durchgefuehrt. Nach dem Abarbeiten des EIEX-Rufes BYE fuer diese Task wird ihre urspruengliche Prioritaet wieder hergestellt.

3.1.3.9. Wechsel der Prioritaet zweier Tasks CHAN

a) Aufgabe: Vertauschen der Prioritaeten zweier Tasks innerhalb der Vorrangorganisation.

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

CHAN TASK = p, CTASK = p [,BYE]

Ruf:

[name] CHAN R=n,(TASK=p,CTASK=p[,BYE]
!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R   – Registerrettung
n   – Registerrettungsart $0 \leq n \leq 2$
TASK – Applikationsprogramm
CTASK– Tauschapplikationsprogramm
p   – Prioritaet der Task $0 < p < 256$
BYE – Prioritaetswechsel bis zur Beendigung der Abarbeitung einer Task
PARA – Parameteradresse
adr – absolute Adresse   0 < adr < FFFFH
symb – symbolische Adresse

d) Ruf-Nr.: 17

e) Ruflaenge: 5 Byte

f) Wirkung: Die angegebenen Tasks tauschen ihre Prioritaeten innerhalb der Vorrangorganisation aus.
Alle Eintragungen in den internen Tabellen des EIEX 1521 werden entsprechend der neuen Prioritaeten aktualisiert. Wird der Parameter BYE im Ruf angegeben, werden beim Beendigen der Abarbeitung einer Task durch den EIEX-Ruf BYE beide Tasks wieder in ihre alte Stellung im Vorrangsystem eingeordnet.
Das Ruecksetzen der Prioritaeten kann zu einem beliebigen Zeitpunkt durch den Ruf CHAN erfolgen (siehe Pkt. 3.1.3.10 Ruecksetzen gewechselter

Prioritaeten zweier Task). Zu beachten ist, dass beim Auslesen von Rufen und Kommandos die Tasks stets mit ihrer urspruenglich vereinbarten Prioritaet anzugeben sind.
Bei erkennbaren Rufparameterfehlern wird der Ruf uebergangen, eine diesbezuegliche Systemnachricht ausgegeben und die Task fortgesetzt.
Die gleiche Reaktion erfolgt ebenfalls, wenn die Task bereits in den zugehoerigen Verwaltungspuffer eingetragen war oder ein Ueberlauf des generierten Verwaltungspuffers auftritt.

3.1.3.10 Ruecksetzen gewechselter Prioritaeten zweier Tasks CHAN

a) Aufgabe: Die gewechselten Prioritaeten zweier Tasks werden auf die urspruenglich vereinbarten Prioritaeten im Vorrangsystem zurueckgestellt

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

```
CHAN    TASK = p
```

Ruf:

```
[name] CHAN   R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TASK	– Applikationsprogramm	
p	– Prioritaet der Task	$0 < p < 256$

PARA - Parameteradresse
adr - absolute Adresse   0 < adr < FFFFH
symb - symbolische Adresse

d) Ruf-Nr.: 17
e) Ruflaenge: 4 Byte
f) Wirkung: Die gewechselten Prioritaeten zweier Tasks, ohne Angabe des Parameters BYE im Ruf, werden auf die urspruenglich festgelegten Prioritaeten zurueckgestellt und die Eintragungen in den internen Tabellen des EIEX 1521 aktualisiert.
Im Ruf CHAN ist dabei nur die Angabe einer Task erforderlich, die zweite zugehoerige Task wird selbstaendig erkannt. Alle Eintragungen im Verwaltungspuffer zur Steuerung des Prioritaetswechsels werden geloescht.
Befindet sich beim Ausloesen des Rufes CHAN die angegebene Task nicht im Verwaltungspuffer, so wird eine Systemnachricht ausgegeben, und die den Ruf ausloesende Task wird fortgesetzt.

3.2. Interruptorganisation

Die Interruptorganisation behandelt die durch die Geraetetechnik des MRS K 1520 ausgeloesten Unterbrechungssignale (Hardware-Interrupt) und die programmtechnischen Unterbrechungen der Abarbeitung von Applikationsprogrammen ueber EIEX-Rufe (Software-Interrupt). Bei allen Interruptgesuchen werden durch die Interruptorganisation die Arbeitsregister nach einer vom Anwender auswaehlbaren Registerrettungsart gerettet und bei der Fortsetzung des unterbrochenen Programmes regeneriert.
Innerhalb eines AP-Systems ist fuer alle Software- und Hardware-

Interruptgesuche nur eine ausgewaehlte Registerrettungsart zulaessig. Jede Task muss einen Stack besitzen. Bei der Dimensionierung des Stacks muessen die Speicherplaetze fuer die gewaehlte Registerrettungsform und die max. anwendungsspezifische Schachtelungstiefe einschliesslich der eventuell erforderlichen Groesse fuer UP-Aufrufe beruecksichtigt werden. Die Stackbelastung betraegt fuer die einzelnen Registerrettungsformen:

- Registerrettungsform 0 — 2 Byte
- Registerrettungsform 1 — 14 Byte
- Registerrettungsform 2 — 22 Byte

EIEX 1521 arbeitet mit einem eigenen Stack, dessen Groesse sich aus der Anzahl der parallel arbeitenden unterbrechbaren Interrupt-Service-Routinen unter Beruecksichtigung der groessten Stackbelastung am jeweiligen Unterbrechungspunkt ergibt.

3.2.1. Beschreibung der Hardware-Interruptbehandlung

Fuer die Behandlung von Hardware-Interruptgesuchen auf Basis der Geraetetechnik des MRS K 1520 ist der Interruptmodus 2 festgelegt, wobei die Abarbeitung von max. 128 Interruptservice-Routinen moeglich ist. In Abhaengigkeit von dem geraeteseitig ausgeloesten Interrupt koennen die Systemkomponenten mit den zugehoerigen Interruptservice-Routinen der

- Zeitorganisation
- Ein-/Ausgabeorganisation fuer periphere Geraete
- Bedienerkommunikation
- Fehlerunterbrechung
- zusaetzlichen Interruptbehandlung zum Start einer speziellen Task

aufgerufen und bedient werden.
EIEX 1521 gewaehrleistet eine Bearbeitung von Interrupt-Verschachtelungen bei unterbrechbaren Interruptservice-Routinen. Die Festlegung ueber einen unterbrechbaren oder nicht unterbrechbaren Ablauf der Interruptservice-Routinen erfolgt entsprechend ihrer Funktion bei der Entwicklung.

3.2.2. Beschreibung der Software-Interruptbehandlung

Die Behandlung der EIEX-Rufe als Software-Interruptgesuche wird ebenfalls durch die Interruptorganisation unterstuetzt, wobei die Einbindung von max. 128 Rufroutinen moeglich ist. In Abhaengigkeit von der Angabe der gewaehlten Registerrettungsart im Ruf werden vor der Ausfuehrung von EIEX-Rufen die entsprechenden Register gerettet und beim Fortsetzen der Abarbeitung der Task regeneriert.

3.3. Organisation des Systemanlaufs

Der Systemanlauf bringt EIEX 1521 in einen definierten Grundzustand. Der Start des Systems erfolgt stets beim Einschalten des Geraetes und/oder beim Druecken der Netztaste, wobei eine generelle Loeschung der CPU (RESET-Funktion) vorangestellt ist. Ein erneuter Systemanlauf kann bei laufendem Betrieb durch Betaetigung einer speziellen Taste mit RESET-Funktion erzielt werden, wobei ebenfalls das Anlaufprogramm gestartet und das EIEX 1521 in den Grundzustand versetzt wird. Auf dem Ausgabegeraet MON 1 erfolgt eine Ausschrift zur Steuerprogrammversion des EIEX 1521. Durch das Anlaufprogramm werden folgende Funktionen ausgefuehrt:

- Alle Systembereiche werden auf Null geloescht. Dabei werden noch vorhandene Anmeldungen fuer die Tasks geloescht und bei erneuter Anmeldung wird deren Bearbeitung bei der Startadresse begonnen. Alle Tasks werden zur Bearbeitung freigegeben.
- Die CPU wird mit dem Interruptmodus 2 , das Interruptregister mit einem generierten Vektor und der Systemstack mit einer Grundadresse geladen.
- Die Geraete der Prozessperipherie und die Tastatur werden initialisiert.
- Die Uhr wird gestartet.

- Die zeitabhaengig zu startenden Programme sind aus der Zeitorganisation ausgetragen.

In dem Anlaufprogramm kann durch Angabe der Prioritaetsnummer eine anwendungsspezifische Anlauftask aufgerufen werden, die weitere Anfangszustaende des Applikationsprogrammsystems herstellt bzw. periphere Geraete initialisiert, die nicht standardmaessig in EIEX 1521 eingebunden sind.
Entfaellt deren Generierung, so befindet sich das Mikrorechnersystem nach Abarbeitung des Anlaufprogramms im dynamischen Stopp Ueber die Kommandoeingabe oder ueber einen externen Interrupt kann dann der Start des Applikationssystem erreicht werden.

3.4. Echtzeituhr und Zeitorganisation

3.4.1. Eigenschaften und Aufbau der Echtzeituhr und des Kalenders

Die Echtzeituhr liefert die Zeitbasis fuer die zeitabhaengige Programmbearbeitung der Tasks und dient als Grundlage des Kalenderprogramms. Die kleinste Zeitbasis wird bei der Generierung festgelegt und als Grundtakt bezeichnet. Er kann 10 ms, 20 ms, 25 ms, 50 ms, 200 ms, 250ms, 500 ms oder 1000 ms betragen.
Das Uhrprogramm behandelt die Interruptgesuche vom Kanal 0 des CTC-Schaltkreises der ZRE K 2521, aktualisiert die Uhrzellen und aktiviert beim Erreichen des jeweiligen Zeitbereiches die Zeitorganisation. Die Zaehlung fuer die einzelnen Zeitbereiche Grundtakt, Sekunde, Minute und Stunde erfolgt getrennt.
Beim Ueberschreiten der 24-Stunden-Grenze wird das Kalenderprogramm gestartet. Es beruecksichtigt die unterschiedliche Anzahl der Tage fuer die einzelnen Monate sowie die Schaltjahre. Tag, Monat und Jahr werden ebenfalls getrennt behandelt.

3.4.2. Organisation der zeitabhaengigen Programme

Mit der Zeitorganisation koennen Tasks zyklisch, verzoegert bzw.

zyklisch und verzoegert gestartet werden. Desweiteren verwaltet sie das zeitabhaengige Warten von Tasks. Diese werden in Abhaengigkeit vom entsprechenden Zeitbereich in vier Zeitebenen eingeordnet. Dabei sind folgende Zeiten moeglich:

- Vielfaches der kleinsten Zeitbasis (max. 32767 mal Grundtakt)
- 1 bis 32767 sec
- 1 bis 32767 min
- 1 bis 32767 hr

Zyklische Tasks starten nach jedem Ablauf der Zykluszeit. Bei verzoegerter Abarbeitung der Tasks erfolgt deren Start nur einmalig. Danach werden die Tasks aus der Zeitorganisation ausgetragen.

Die Verzoegerungszeit kann dabei als Startzeit (z.B. 14.30 Uhr) oder als Startzeitdifferenz (z.B. 10 Minuten) eingegeben werden. Zykluszeit, Startzeitdifferenz und Pausezeit werden stets als ganze positive Zahlen in Verbindung mit den entsprechenden Zeitbereichen eingegeben. Die Angabe der Startzeit erfolgt durch Punkt getrennt in Stunden und Minuten.

Beim verzoegerten, zyklischen Start von Tasks sind unterschiedliche Zeitbereiche fuer die Verzoegerungszeit und Zykluszeit zulaessig, jedoch benoetigen sie den doppelten Speicherplatz und mehr Verwaltungszeit. Besitzen mehrere Tasks einen gleichen Startzeitpunkt, so startet das Vorrangsystem die Task mit der hoechsten Prioritaet.

3.4.3. Besonderheiten der Echtzeituhr und Zeitorganisation

- Die Eingabe der aktuellen Uhrzeit und des aktuellen Datums erfolgt nach Systemanlauf mittels der Kommandos STIME und SDATE durch den Anwender, so dass eine Modifikation dieser Daten moeglich ist. Grundtakt und Sekunde werden bei Eingabe der Uhrzeit generell gleich Null gesetzt.

- Die Eingabe von Zyklus- und Verzoegerungszeit gleich Null ist unzulaessig.
- Die Anzahl der in den Zeitebenen eintragbaren Programme ist beliebig, sie darf jedoch die Gesamtzahl des bei der Generierung festgelegten Wertes nicht ueberschreiten. Maximal koennen 50 Tasks zeitlich verwaltet werden.
- Jede Zeitebene wird einzeln verwaltet.
  So erfolgt z.B. der Start von Tasks im Stundenbereich nur zur vollen Stunde. Soll eine Task z.B. zu jeder halben Stunde gestartet werden, ist eine Zykluszeit von 60 Minuten einzugeben und die Verzoegerungszeit so zu waehlen, dass der erste Startzeitpunkt genau bei 30 Minuten liegt. Soll eine Task z.B. um 14.30 Uhr zyklisch alle 30 Minuten gestartet werden, sind folgende Startvarianten moeglich:
  a) Eingabe der Zykluszeit (30 min) genau um 14.30 Uhr.
  b) Eingabe der Startzeit (14.30 Uhr) und der Zykluszeit (30 min)
  c) Eingabe der Startzeitdifferenz (z.B. bei einer Uhrzeit von 14.05 Uhr wuerde sie 25 Minuten betragen) und der Zykluszeit (30 min).

3.4.4. EIEX-Rufe und Kommandos

3.4.4.1. Verzoegerter Start einer Task RUN

a) Aufgabe: Einmaliger verzoegerter Start einer Task
b) Status: Ruf oder Kommando
c) Schreibweise:

Kommando:

```
RUN  TASK = p, (DET=t(HR!MI!SE!CL)!RT=hr.mi)
```

Ruf:

```
[name] RUN  R=n,(TASK=p,(DET=t(HR!MI!SE!CL)!RT=hr.mi)
              !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TASK	– Applikationsprogramm	
p	– Prioritaet einer Task	$0 < p < 256$
DET	– Verzoegerungszeit	
RT	– Absolute Startzeit	
t	– Wert der Verzoegerungszeit	$0 < t < 32768$
hr.mi	– Wert der absoluten Startzeit	$0 \leq hr \leq 23$ $0 \leq mi \leq 59$
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr.: 0

e) Ruflaenge: 7 Byte

f) Wirkung: Die Task wird nach Ablauf einer vorgegebenen Zeit oder zur gewuenschten Startzeit in der Vorrangorganisation angemeldet und entsprechend ihrer Prioritaet gestartet.
Wird die zu startende Task bereits zeitlich verwaltet, so erfolgt die Ausgabe einer Fehlerinformation auf dem Protokollgeraet und die den Ruf ausloesende Task wird fortgesetzt. Die gleiche Reaktion erfolgt bei Ueberschreitung der Anzahl der generierten zeitabhaengigen Tasks sowie bei dem Startversuch einer verhinderten Task.

3.4.4.2. Zyklischer Start einer Task RUN

a) Aufgabe: Sofortiger zyklischer Start einer Task

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando: RUN TASK = p, CT = t(HR!MI!SE!CL)

Ruf:

[name] RUN R=n,(TASK=p,CT=t(HR!MI!SE!CL)
!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
TASK	- Applikationsprogramm	
p	- Prioritaet einer Task	$0 < p < 256$
CT	- Zykluszeit	
t	- Wert der Zykluszeit	$0 < t < 32768$
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr.: 0

e) Ruflaenge: 7 Byte

f) Wirkung: Die Task wird sofort in der Vorrangorganisation angemeldet und entsprechend ihrer Prioritaet gestartet. Die weiteren Starts erfolgen stets nach Ablauf der Zykluszeit.
Eine fehlerhafte Benutzung des Rufes hat die gleichen Reaktionen analog Pkt. 3.4.4.1, Verzoegerter Start einer Task, zur Folge.

3.4.4.3. Verzoegerter zyklischer Start einer Task RUN

a) Aufgabe: Verzoegerter zyklischer Start einer Task

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

RUN TASK=p,(DET=t(HR!MI!SE!CL)!RT=hr.mi),CT=t(HR!MI!SE!CL)

Ruf:

```
[name] RUN  R=n,(TASK=p,(DET=t(HR!MI!SE!CL)!RT=hr.mi),
                 CT=t(HR!MI!SE!CL)
                !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TASK	– Applikationsprogramm	
p	– Prioritaet einer Task	$0 < p < 256$
DET	– Verzoegerungszeit	
RT	– absolute Startzeit	
CT	– Zykluszeit	
t	– Wert der Verzoegerungs- bzw. Zykluszeit	$0 < t < 32768$
hr.mi	– Wert der Startzeit	$0 \leq hr \leq 23$ $0 \leq mi \leq 59$
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr.: 0

e) Ruflaenge: 9 Byte

f) Wirkung: Die Task wird nach Ablauf der Verzoegerungszeit bzw. zur Startzeit in der Vorrangorganisation angemeldet und entsprechend ihrer Prioritaet gestartet. Die weiteren Starts erfolgen jeweils nach Ablauf der Zykluszeit.
Eine fehlerhafte Benutzung des Rufes hat die gleichen Reaktionen, analog Pkt. 3.4.4.1, Verzoegerter Start einer Task, zur Folge.

3.4.4.4. Zeitliche Unterbrechung einer Task PAUS

a) Aufgabe: Unterbrechung der Abarbeitung der aktuellen Task bis zum Ablauf einer Pausezeit

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name]  PAUS R=n,(PTIM=t(HR!MI!SE!CL)[,PRT]
                    !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R – Registerrettung

n – Registerrettungsart $0 \leq n \leq 2$

PTIM – Pausezeit

t – Wert der Pausezeit $0 < t < 32768$

PRT – Protokollierung des Pausezustandes

PARA – Parameteradresse

adr – absolute Parameteradresse

symb – symbolische Adresse

d) Ruf-Nr.: 3

e) Ruflaenge: 6 Byte

f) Wirkung: Der Ruf setzt die aktuelle Task bis zum Ablauf der vorgegebenen Zeit in den Wartezustand. Der Pausezustand kann auf dem Bediengeraet protokolliert werden. Eine zeitlich pausierende Task kann nicht mit dem Ruf GO fortgesetzt werden. Die fehlerhafte Benutzung des Rufes hat die gleichen Reaktionen, analog Pkt. 3.4.4.1, Verzoegerter Start einer Task, zur Folge.

3.4.4.5. Austragen der Tasks aus der Zeitorganisation CNCL

a) Aufgabe: Austragen einer oder aller Task aus der Zeitorganisation

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

CNCL (TASK=ALL!TASK=p),TM

Ruf:

[name] CNCL R=n,(ALL!TASK=p),TM
!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
TASK	- Applikationsprogramm	
p	- Prioritaet der Task	0 < p < 256
TM	- Austragen einer Task aus der Zeitorganisation	
ALL	- Austragen aller Tasks aus der Zeitorganisation	
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr.: 2

e) Ruflaenge:

bei Angabe der Task: 4 Byte

ohne Angabe der Task: 3 Byte

f) Wirkung: Die angegebene Task in Verbindung mit dem Schluesselwort TM wird aus der Liste der zeitabhaengigen Programme ausgetragen und demzufolge nicht mehr zeitlich verwaltet.
Bei Angabe von CNCL TASK=ALL werden alle Tasks aus der Zeitorganisation ausgetragen und demzufolge nicht mehr zeitlich verwaltet.

3.4.4.6. Stellen der Echtzeituhr STIME

a) Aufgabe: Aktualisieren der Uhrzeit

b) Status: Kommando

c) Schreibweise:
   Kommando:
   STIME hr.mi

   Dabei bedeuten die Schluesselworte und Parameter:
   hr – Aktuelle Stunde $0 \leq hr \leq 23$
   mi – Aktuelle Minute $0 \leq mi \leq 59$

f) Wirkung: Die Echtzeituhr wird mit den eingegebenen Zeitwerten aktualisiert, die Sekunden werden auf Null gesetzt.

3.4.4.7. Abrufen der Echtzeituhr — TIME

a) Aufgabe: Bereitstellen der Uhrzeit im Speicher
b) Status: Ruf
c) Schreibweise:
   Ruf:
   [name] TIME R=n,(BOB=(adr!symb)!PARA=(adr!symb))[;Kommentar]

   Dabei bedeuten die Schluesselworte und Parameter:
   R – Registerrettung
   n – Registerrettungsart $0 \leq n \leq 2$
   BOB – Zieladresse
   PARA – Parameteradresse
   adr – absolute Adresse $0 < adr < FFFFH$
   symb – symbolische Adresse

d) Ruf-Nr: 12
e) Ruf-Laenge: 5 Byte
f) Wirkung: Beginnend ab der angegebenen Zieladresse werden die internen Zeitwerte der Echtzeituhr durch Doppelpunkt getrennt achtstellig im ISO-Code abgespeichert (Form: HR:MI:SE).

3.4.4.8. Stellen des Kalenders — SDATE

a) Aufgabe: Aktualisieren des Datums

b) Status: Kommando

c) Schreibweise:

Kommando:

SDATE da.mo.ye.

Dabei bedeuten die Schluesselworte und Parameter:

da	- Aktuelles Tagesdatum	$0 < da < 32$
mo	- Aktueller Monat	$0 < mo < 13$
ye	- Aktuelles Jahr	$0 \leq ye \leq 99$

f) Wirkung: Der Kalender wird mit den eingegebenen Werten aktualisiert.

3.4.4.9. Abrufen des Datums DATE

a) Aufgabe: Bereitstellen des Datums im Speicher

b) Status: Ruf

c) Schreibweise:

Ruf:

[name] DATE R=n,(BOB=(adr!symb)!PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
BOB	- Zieladresse	
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr: 13

e) Ruf-Laenge: 5 Byte

f) Wirkung: Beginnend mit der angegebenen Zieladresse werden die internen Datumswerte durch Punkt getrennt achtstellig im ISO-Code abgespeichert (Form: DA.MO.YE).

3.5. Organisation der Unterprogramme

3.5.1. Moeglichkeiten der UP-Organisation

Die Unterprogrammorganisation gewaehrleistet die Arbeit mit folgenden Arten von Unterprogrammen:

- anwendereigene UP
- durch Interrupt unterbrechbare Bibliotheks - UP
- durch Interrupt nicht unterbrechbare Bibliotheks - UP

Die anwendereigenen UP gehoeren funktionell zur Task und sind nicht Bestandteil der UP-Bibliothek.
Die zur Mehrfachbenutzung bestimmten UP werden in einer Unterprogrammbibliothek zusammengefasst.
Auf die Programme der UP-Bibliothek haben alle Tasks Zugriff.
Bei der Arbeit mit unterbrechbaren Unterprogrammen verwaltet die UP-Organisation die Reihenfolge ihrer Benutzung durch die Tasks unter voller Beachtung ihrer Stellung im Vorrangsystem. Dadurch wird die Mehrfachbenutzung durch Tasks unterschiedlicher Prioritaet in ihrer zeitlichen Abarbeitung gesteuert.
Die unterbrechbaren und die anwendereigenen Unterprogramme besitzen eine beliebige Anzahl von Ein- und Austrittspunkten.
Die nicht unterbrechbaren Unterprogramme muessen kurze, konfigurationsabhaengige Abarbeitungszeiten besitzen. Bei Anschluss eines Folienspeichers duerfen sie nicht ueber 0,600 ms, bei jeglicher anderer DV-Peripherie nicht ueber 1 ms liegen.
Sie duerfen keine unterbrechbaren Unterprogramme aufrufen.

3.5.2. Benutzungshinweise zur UP-Organisation

Die Anzahl der unterbrechbaren Unterprogramme muss bei der Generierung festgelegt werden.
Bibliotheksunterprogramme duerfen keine EIEX-Rufe enthalten und nur UP's der UP-Bibliothek aufrufen.
Der symbolische Name eines Bibliotheksunterprogramms ist stets mit zwei alphanumerischen Zeichen anzugeben. Der Name des Eintrittspunktes kann zwischen 2 und 5 Zeichen variieren.

Die UP-Bibliothek verwaltet die richtige Reihenfolge der UP-Benutzung durch Tasks mit unterschiedlicher Prioritaet.
Ein UP der Bibliothek kann erst dann wieder gestartet werden, wenn die vorhergehende Bearbeitung abgeschlossen ist. Demzufolge kann eine Task hoeherer Prioritaet erst auf ein zur Zeit arbeitendes UP zugreifen, wenn die Task mit niedrigerer Prioritaet das UP verlaesst. Dabei erhaelt die UP belegende Task bis zum Austritt aus dem UP, die Prioritaet der UP-rufenden Task.
Saemtliche Bibliotheksunterprogramme koennen wie herkoemmliche UP's geschrieben werden und enden demzufolge immer mit dem RET-Befehl bzw mit einem bedingten Ruecksprung (z.B. RZ-Befehl).
Die Einbindung von UP's in die Bibliothek reduziert den Speicherbedarf, erhoeht aber bei unterbrechbaren UP's, bedingt durch die UP-Verwaltung, die Abarbeitungszeit eines UP's.
Unterbrechbare UP's sollten daher zeitunkritisch sein. Eine Parameteruebergabe in den Registern kann beim Aufruf und Verlassen des UP's erfolgen. Bei der Arbeit mit Folienspeichern fuehrt die Benutzung der Tauschregister zu fehlerhaften Ablaeufen und ist nicht gestattet.

3.5.3. Rufe

3.5.3.1. Aufruf eines nichtunterbrechbaren Bibliotheksunterprogramms LISD

a) Aufgabe: Aufruf eines nichtunterbrechbaren UP's aus der UP-Bibliothek
b) Status: Ruf
c) Schreibweise:
Ruf:
[name] LISD LIB=SR.EP[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:
SR – Symbolischer Name des Unterprogramms

EP - Symbolischer Name des Eintrittspunktes

d) Wirkung: Das UP wird unter seinem symbolischen Namen aufgerufen und ohne Unterbrechung abgearbeitet

3.5.3.2. Aufruf eines unterbrechbaren Bibliotheksunterprogramms LISE

a) Aufgabe: Aufruf eines unterbrechbaren UP's aus der UP-Bibliothek

b) Status: Ruf

c) Schreibweise:

Ruf:

[name] LISE LIB=SR.EP[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

SR - Symbolischer Name des Unterprogramms

EP - Symbolischer Name des Eintrittspunktes

d) Wirkung: Das UP wird unter seinem symbolischen Namen aufgerufen und abgearbeitet. Unterbrechungen sind moeglich.

3.6. Ein-/Ausgabeorganisation fuer Geraete der Datenverarbeitungsperipherie

Die Programmoduln der Ein-/Ausgabeorganisation fuer Geraete der Datenverarbeitungsperipherie sind generierbare Bestandteile des EIEX 1521.
Die DV-Organisation uebernimmt den organisatorischen Ablauf der Datenuebertragungen zwischen Arbeitsspeicher und Datenverarbeitungsperipherie.
Die EIEX-Rufe zur Datenuebertragung koennen in den Tasks des Applikationsprogrammsystems oder in den dazugehoerigen applikationseigenen Unterprogrammen programmiert werden.
Die DV-Organisation ermoeglicht unter EIEX 1521 die parallele Arbeitsweise von verschiedenen peripheren Geraeten und der Zentraleinheit (und damit einer Task).
Die DV-Organisation ist modular aufgebaut und besteht aus folgenden Funktionskomponenten:
- Modul fuer die zentrale Rahmensteuerung und Warteschlangenbearbeitung
- Modul fuer die zeitliche Ueberwachung von peripheren Geraeten
- Modul fuer eine automatische Geraeteumschaltung

Unmittelbar mit der DV-Organisation sind die Steuermoduln fuer die Geraeteum- und Geraeterueckschaltung, sowie die Treiber und Interruptservice-Routinen fuer die peripheren Geraete verbunden. Sie realisieren unmittelbar die Datenuebertragung zwischen dem Arbeitsspeicher und dem peripheren Geraet.

3.6.1. Ein/Ausgabeorganisation fuer Applikationsprogramme

Unter EIEX 1521 stehen dem Nutzer fuer die Arbeit mit der DV-Peripherie folgende Rufe und Kommandos zur Verfuegung:
a) Rufe:
COTR - Steuern und Positionieren eines Geraetes

READ - Eingabe der Daten von Datentraegern
WRIT - Ausgabe der Daten auf Datentraeger
WAIT - Test auf Beendigung eines E/A-Vorganges
ASGN - Geraetezuweisung, Geraeteum- und Geraeterueckschaltung

b) Kommando:
ASGN - Geraetezuweisung, Geraeteum- und Geraeterueckschaltung

Die Ein- oder Ausgabe von Daten wird in einer Task oder in einem applikationseigenen Unterprogramm durch die EIEX-Rufe READ, WRIT und COTR eingeleitet. Dabei wird in die zentrale Rahmensteuerung der DV-Organisation verzweigt.
Die DV-Organisation fuehrt die Pruefung der im Ruf enthaltenen logischen Geraete-Nummer sowie eine Kontrolle zur Verfuegbarkeit des logischen Geraetes fuer das Eroeffnen der Ein- und/oder Ausgabe durch.
Kann der ausgeloeste E/A-Ruf auf dem DV-Geraet abgearbeitet werden, so wird durch den zugehoerigen Geraetetreiber die Ein- bzw. Ausgabe von Daten eingeleitet. Bei einer im E/A-Ruf angegebenen WAIT-Bedingung wird die Task fuer die Zeitdauer der vollstaendigen Bearbeitung der Ein- bzw. Ausgabe in einen Wartezustand gesetzt. Ist diese nicht im E/A-Ruf enthalten, so erfolgt eine Registerregenerierung durch die Interruptorganisation und die Task-Bearbeitung wird fortgesetzt.
Die Datenuebertragung verlaeuft losgeloest von der Abarbeitung der zugehoerigen Task. Nach Beendigung der Datenuebertragung erfolgt die weitere Bearbeitung der Task unter Auswertung der WAIT-Bedingung (siehe Pkt. 3.6.4.6, Test auf Ende eines E/A-Rufes). Zur Kontrolle der ordnungsgemaess durchgefuehrten Datenuebertragung ist aus Sicherheitsgruenden in jedem E/A-Ruf eine Adresse fuer einen Fehlerschluessel anzugeben, unter der EIEX 1521 eine Information (Fehlercode) ueber den Verlauf der Datenuebertragung bereitstellt. Der abgespeicherte Code ist durch die Task auszuwerten, um fehlerhafte Datenuebertragungen erkennen und behandeln zu koennen.
Eine moegliche Reaktion der Task auf bestimmte Fehlerschluessel

bei der Datenuebertragung besteht in der Umschaltung eines DV-Geraetes auf ein anderes Peripheriegeraet mittels des EIEX-Rufes ASGN oder im Aufruf eines applikationsspezifischen Fehlermassnahmeprogramms. Bei bestimmten Geraetefehlern, die in einigen Geraetetreibern erkannt werden, kann eine automatische Geraeteumschaltung auf ein verfuegbares Ersatzgeraet gleichen Geraetetyps wirksam werden.
Durch den EIEX-Ruf ASGN ist ausserdem eine Geraetezuweisung moeglich, wobei sich dann die generierte Zuordnung zwischen logischer Geraetenummer und physischem Geraet veraendert.
Bei der Generierung kann jedem DV-Geraet eine Warteschlange zugeordnet werden. In die Warteschlange werden, falls das DV-Geraet beim Ausloesen eines E/A-Rufes nicht verfuegbar ist, alle Ein- und Ausgabeanforderungen eingetragen und entsprechend ihrer Prioritaet nach Beendigung einer Datenuebertragung durch EIEX 1521 erneut gestartet. Bei der Arbeit mit den Folienspeichern ist die Benutzung der Tauschregister nicht gestattet.

3.6.2. Ein-/Ausgabeorganisation fuer Bediener

Durch Kommandoeingabe auf der Tastatur kann der Bediener auf die Arbeit des Echtzeitsteuerprogrammsystem EIEX 1521 Einfluss nehmen. Die Kommandos zur Geraeteum- bzw. Geraeterueckschaltung sowie zur Geraetezuweisung besitzen eine steuernde Wirkung auf die Arbeitsweise der DV-Peripherie. Treten bei der Abarbeitung von E/A-Rufen fehlerhafte Zustaende auf, so werden relevante Informationen als Systemnachrichten dem Bediener zur Auswertung uebergeben. Die Systemnachrichten geben Auskunft ueber den Betriebszustand des DV-Geraetes und fordern gegebenenfalls zur Reaktion durch den Bediener auf. Die ausgegebenen Systemnachrichten fuer die DV-Geraete sind durch den Bediener, unter Verwendung der Anlage 3 (Fehlerausschriften und Systemnachrichten EIEX 1521), besonders sorgfaeltig zu analysieren.

3.6.3. Funktionen der E/A-Organisation

Die zentrale Rahmensteuerung bildet das Kernstueck der E/A-Organisation, in der die zentralen Steuerfunktionen integriert

sind. Sie beinhalten allgemeingueltige Kontroll- und Verwaltungsablaeufe fuer die Geraete der DV-Peripherie unter Beruecksichtigung der Vorrangorganisation.
Weiterhin ist eine Speicherverwaltung fuer den Anwenderbereich der Bildschirmbaugruppe MON1 enthalten. Sie gewaehrleistet bei Tastatureingaben in den Nutzerbereich des MON1 einen temporaeren Schutz gegenueber einem parallelen Speicherzugriff auf gleiche Adressraeume bei Ausgaben auf den Nutzerbereich der Bildschirmbaugruppe.
Da eine Arbeitsweise der Geraete der DV-Peripherie ohne Warteschlangen moeglich ist, muss durch die E/A-Organisation die Verwaltung der E/A-Rufe in Verbindung mit der Vorrangorganisation gesteuert werden. In dieser Betriebsform kann eine zeitweilig starke Belastung des Echtzeitsteuerprogrammsystems auftreten. Weiterhin wird die Zusammenarbeit mit dem Handler zur Dateiarbeit auf Folienspeicher gesteuert.

3.6.3.1. Warteschlangen-Organisation

Jedes DV-Geraet besitzt eine Warteschlange von konstanter Laenge, in der alle nicht ausloesbaren E/A-Rufe verwaltet werden. In die Warteschlange werden im allgemeinen alle Tasks, deren E/A-Rufe durch den Besetztzustand des DV-Geraetes nicht eroeffnet werden koennen, eingeordnet und gleichzeitig in einen Wartezustand gesetzt.
Wird das Ende einer laufenden Datenuebertragung fuer das Geraet erreicht, so erfolgt eine Kontrolle auf aktuelle Eintragungen in der Warteschlange. Sind mehrere Eintragungen fuer das DV-Geraet vorhanden, so wird die Task mit der hoechsten Prioritaet unter Beachtung ihres Steuerzustandes zur Bearbeitung angemeldet. Diese Task besetzt dann das DV-Geraet und fuehrt entsprechend der Parameter seines E/A-Rufes die Datenuebertragung durch. Aus der dargestellten Arbeitsweise ist ersichtlich, dass die Arbeit der DV-Geraete ebenfalls nach Prioritaeten entsprechend des absoluten Vorrangs fuer Tasks gesteuert wird.

3.6.3.2. Geraeteumschaltung, Geraeterueckschaltung und Geraetezuweisung

Die E/A- Rufe READ, WRIT und COTR enthalten stets eine logische Geraetenummer. Diese bleibt bei den nachfolgend beschriebenen Geraetemanipulationen unveraendert erhalten. Die Geraeteumschaltung, Geraeterueckschaltung oder Geraetezuweisung beeinflusst lediglich die Zuordnung zwischen der im E/A- Ruf angegebenen logischen Geraetenummer und dem physischen Geraet. Bei jedem Systemanlauf von EIEX 1521 wird die generierte Zuordnung zwischen logischer Geraetenummer und physischem Geraet hergestellt, so dass dabei vorangegangene Geraetezuweisungen bzw. Geraeteumschaltungen nicht erhalten bleiben.

a) Geraeteumschaltung

Eine Geraeteumschaltung ist nur zwischen DV- Ausgabegeraeten gleichen Typs moeglich (LBS bzw. SD). Sie ermoeglicht, eine fehlerhaft verlaufene E/A- Operation (Geraetestoerung) auf einem zugewiesenen Ersatzgeraet erneut zu starten. Dabei wird das angegebene defekte Geraete auf Stoerung und auf ein verfuegbares Ersatzgeraet gleichen Typs geprueft. Die Geraeteumschaltung wird sofort ausgefuehrt, wenn auf dem Ersatzgeraet ab Systemstart keine Daten ausgegeben worden sind. Hat das Geraet bereits gearbeitet, erfolgt keine Geraeteumschaltung, da eine Ausgabe von unterschiedlichen Daten auf einem Datentraeger (Datenmix) zu erwarten ist. Kann dieser Fall im Applikationsprogrammsystem ausgeschlossen werden, so wird beim erneuten Aufruf der Geraeteumschaltfunktion per Ruf oder Kommando die Geraeteumschaltung zwangsweise durchgefuehrt.
Nach erfolgter Geraeteumschaltung werden automatisch der fehlerhaft verlaufene E/A- Ruf sowie alle kuenftigen E/A-Operationen auf dem Ersatzgeraet abgearbeitet.

b) Automatische Geraeteumschaltung

Bei einer generierten automatischen Geraeteumschaltung fuer DV-Geraete zur Datenausgabe wird beim Erkennen von bestimmten Hardware- Fehlern (Statusmeldungen) im Geraetetreiber die Geraeteumschaltung automatisch eroeffnet, wenn gleichzeitig folgende Bedingungen erfuellt sind:

- dem defekten Geraet wurde bei der Generierung ein Ersatzgeraet zugeordnet
- das Ersatzgeraet ist nicht bereits selbst auf das ausgefallene Geraet umgeschaltet.

Dabei wird primaer versucht, das defekte Geraet durch ein Ersatzgeraet zu substituieren, auf das nach Systemanlauf noch keine Daten ausgegeben wurden. Waren bereits alle vorgesehenen Ersatzgeraete aktiv, so erfolgt die Umschaltung auf ein zur Zeit nicht arbeitendes Geraet. Durch entsprechende Systemnachrichten wird auf die durchgefuehrte Geraeteumschaltung sowie auf die Ausgabe evtl. unterschiedlicher Daten auf einem Datentraeger hingewiesen. Der durch einen Geraetefehler unterbrochene E/A-Ruf wird automatisch auf dem Ersatzgeraet wiederholt. Danach werden die in der Warteschlange oder Vorrangorganisation verwalteten sowie alle zukuenftigen E/A- Rufe auf dem Ersatzgeraet ausgefuehrt.

c) Geraeterueckschaltung

Durch die Geraeterueckschaltung kann ein repariertes oder ausgetauschtes Geraet der DV- Peripherie wieder unter dem bei der Generierung festgelegten logischen und physischen Geraet in den Echtzeitbetrieb eingeordnet werden. Die Geraeterueckschaltung wird nur fuer gestoerte Geraete durchgefuehrt, sonst erfolgt eine Systemnachricht und die Funktion wird nicht ausgefuehrt. Eine

laufende E/A- Operation wird durch die Geraeterueckschaltung nicht beeinflusst. Alle bereits durch die Warteschlange oder Vorrangorganisation verwalteten sowie die zukuenftigen E/A- Rufe werden auf dem rueckgeschalteten Geraet ausgefuehrt.

d) Geraetezuweisung

Die Geraetezuweisung vertauscht zwischen zwei Geraeten die generierte Beziehung von logischem und physischem Geraet. Sie wird nur dann ausgefuehrt, wenn die angegebenen logischen Geraete vom gleichen Typ sind und zum Zeitpunkt der Ausfuehrung der Geraetezuweisung nicht gestoert sind bzw. aktiv arbeiten. Werden diese Voraussetzungen nicht erfuellt, folgt eine diesbezuegliche Systemnachricht. Es wird ein Fehlercode zur Auswertung bereitgestellt und die Geraetezuweisung nicht ausgefuehrt.

3.6.3.3. Zeitueberwachung der E/A- Operation

EIEX 1521 ueberwacht den Datenaustausch von bzw. zu den Geraeten der DV- Peripherie, damit auftretende Geraetefehler schnell erkannt und geeignete Massnahmen zu ihrer Behebung eingeleitet werden. Die zeitliche Ueberwachung erfolgt in zwei Formen:

a) Autonome Zeitueberwachung durch ein zentrales Zeitkontrollprogramm

In EIEX 1521 erfolgt diese Form der Ueberwachung fuer DV- Geraete bei einer zeichen- oder blockweisen Uebertragung von Daten. Dabei wird vor Beginn jeder Datenuebertragung in einer dem Zeitkontrollprogramm zugeordneten Tabelle eine fuer das spezifische DV- Geraet relevante Zeitkonstante eingetragen. Die Groesse der Zeitkonstanten ist fuer jedes zu ueberwachende DV- Geraet so festgelegt, dass im fehlerfreien Betrieb die Datenuebertragung

stets vor dem Erreichen des Zeitwertes Null ordnungsgemaess abgeschlossen ist. Das Zeitkontrollprogramm arbeitet nach einem generierbaren Zeitzyklus die eingetragenen Werte in der Zeittabelle ab und registriert beim Nulldurchgang eine Zeitueberschreitung fuer das DV- Geraet. In diesem Fall wird eine diesbezuegliche Systemmeldung ausgegeben sowie ein Fehlerschluessel bereitgestellt, der durch die Task auszuwerten ist. Diese Form der Zeitueberwachung wird fuer die DV- Gerate Seriendrucker, Lochbandleser und Lochbandstanzer verwendet.

b) Treiberinterne Zeitueberwachung

Die treiberinterne Zeitueberwachung wird fuer die Ueberwachung von DV- Geraeten mit mehreren unterschiedlichen Zeitwerten, die eine geringe Toleranz im Wertebereich besitzen, angewendet. So erfolgt in EIEX 1521 die Ueberwachung der Arbeit der Folienspeicherlaufwerke unter dem CTC- Kanal 1. Dabei wird im Treiber die Zeitkontrolle parallel zu allen kritischen Ablaeufen aktiviert und somit die zeitlich zu erwartenden Folgereaktionen ueberwacht. Bleibt die zu erwartende Reaktion aus oder erfolgt sie zu spaet, dann wird ein treiberinternes Fehlermassnahmeprogramm abgearbeitet und die E/A- Operation bis zu einer definierten Abbruchbedingung wiederholt. Kann der vorliegende Fehler vom Treiber nicht behoben werden, dann erfolgt die Ausgabe einer Systemnachricht mit Bereitstellung eines diesbezueglichen Fehlerschluessels, der durch die Task auszuwerten ist.

3.6.4. Rufe und Kommandos

3.6.4.1. Benutzungshinweise

Die Ein- und Ausgabe von Daten erfolgt bei den Geraeten der DV-Peripherie innerhalb des Applikationsprogrammsystems allgemein als Datenblock. Dabei wird die logische Blockgroesse durch den

Anwender selbst definiert und stellt eine frei waehlbare Menge von Bytes dar. In allen E/A- Rufen wird die zu verarbeitende Datenmenge als Block aufgefasst, der durch eine Pufferanfangsadresse und Pufferlaenge gekennzeichnet ist. Die Adressangabe bezieht sich auf den Arbeitsspeicher, der vom Datenblock belegt wird oder belegt werden soll. In Abhaengigkeit vom DV- Geraet erfolgt durch den Geraetetreiber eine Transformation des logischen Datenblocks auf die Groesse der physischen Datenbloecke und deren wiederholte Ein- oder Ausgabe aus geraetespezifischen Arbeitsspeichern.
Eine Blockuebertragung ist regulaer beendet, wenn die Zeichenanzahl ausgegeben ist, die auch durch ein spezifisches Endekennzeichen abgeschlossen werden kann.
In jedem E/A-Ruf ist die Adresse einer dem Ruf zugehoerigen Fehlerzelle anzugeben, in die nach Beendigung der E/A-Operation evtl. ein Fehlerschluessel eingetragen wird. Der Inhalt der Fehlerzelle ist nach jedem E/A-Ruf oder nach dem EIEX-Ruf WAIT in der Task auszuwerten, um gegebenenfalls die notwendigen Fehlermassnahmen programmtechnisch ausloesen zu koennen.
Erfolgt die Programmierung eines E/A-Rufes ohne WAIT-Bedingung, so kann durch den EIEX-Ruf WAIT das Ende der Datenuebertragung abgetestet werden.
Besteht bei der Ein- oder Ausgabe von Daten die Notwendigkeit eines Codewandels, so kann dies durch den EIEX-Rufes CODE realisiert werden. Der Anwender legt dabei den Inhalt der Codetabellen unter Beachtung der Hinweise unter Punkt 3.10.3, Codewandlung, selbst fest.

3.6.4.2. Steuern und Positionieren

3.6.4.2.1. COTR fuer SD 1156 COTR

a) Aufgabe: Dieser Ruf fuehrt ausschliesslich Steuerfunktionen fuer den Seriendrucker 1156 durch

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] COTR  R=n,(LP=i(,BPOS=d!(FLF=(s!z),LEP=(1!2!3)))
                 [,ECA=(adr!symb)][,WAIT]
                 ! PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
LP	– Seriendrucker	
i	– logische Geraetenummer	$0 < i < 256$
BPOS	– Anfangsangaben zur Position	
d	– Position des ersten Druckzeichens	$0 < d < 171$
FLF	– Angaben zur Formulartraegersteuerung	
s	– einmalige Seitenschaltung	
z	– Zeilenschaltung	$0 < z < 32$
LEP	– Formulartraegersteuerung	
1	– Leporello 1	
2	– Leporello 2	
3	– Leporello 1 und 2	
ECA	– Fehlerschluesseladresse	
WAIT	– Warten auf E/A-Ende	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 8-10 Byte

f) Wirkung: Entsprechend den im Ruf spezifizierten Parametern werden entweder der Druckkopf auf die angegebene Stelle positioniert oder ein Transport der angegebenen Formulartraeger als Zeilen- oder Seitenvorschub fuer Leporello 1 und/oder 2 durchgefuehrt.

g) Fehler:

Fehlercode=10H: Im Ruf sind fehlerhafte Parameter enthalten. Es wird der Ruf uebergangen und die Task fortgesetzt.

Fehlercode=40H: Statusfehler (Papierende, rechter oder linker Rand). Der Ruf wird uebergangen und die Task fortgesetzt. Ein Bedienereingriff ist erforderlich.

3.6.4.2.2. COTR fuer LBS 1215 COTR

a) Aufgabe: Dieser Ruf fuehrt ausschliesslich Steuerfunktionen fuer den Lochbandstanzer 1215 durch

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] COTR  R=n,(PTP=i,IN=k[,ECA=(adr!symb)][,WAIT]
                   !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
PTP	– Lochbandstanzer	
i	– logische Geraetenummer	$0 < i < 256$
IN	– Geraetekommando	
k	– Nummer des Geraetekommandos	
ECA	– Fehlerschluesseladresse	
WAIT	– Warten auf E/A-Ende	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 6-8 Byte

f) Wirkung: Entsprechend dem im Ruf angegebenen Geraetekommando wird das Lochband ausgetrieben oder ein Rueckschritt ausgefuehrt. Dabei haben die einzelnen Geraetekommandos folgende Funktionen:

k = 1: Lochbandaustrieb (ein Zeichen)

k = 2: Lochbandrueckschritt (ein Zeichen)

g) Fehler:

Fehlercode=10H: Im Ruf sind fehlerhafte Parameter enthalten. Es wird der Ruf uebergangen und die Task fortgesetzt.

Fehlercode:40H: Datentraegerriss. Es wird der Ruf abgebrochen und die Task fortgesetzt.

Fehlercode=80H: Datentraegerendevorwarnung. Die Abarbeitung des Rufes wird durch diesen Fehler nicht beeinflusst.

3.6.4.2.3. COTR fuer Bildschirmbaugruppe MON1 COTR

a) Aufgabe: Dieser Ruf loescht teilweise oder komplett die im generierten Nutzerbereich gespeicherten Informationen.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] COTR  R=n,(MON=i,IN=k,BPOS=z/s(,LOB=j!EPOS=z/s)
                   [,ECA=(adr!symb)][,WAIT]
                   !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
MON	– Bildschirmausgabebaugruppe	
i	– logische Geraetenummer	$0 < i < 256$
IN	– Geraetekommando	
k	– Nummer des Geraetekommandos	

BPOS	– Anfangsangaben zur Position	
EPOS	– Endangaben zur Position	
z	– Zeile	0 < z < 17
s	– Spalte	0 < s < 65
LOB	– Bereichslaenge	
j	– Anzahl der zu loeschenden Zeichen	0 < j < 1025
ECA	– Fehlerschluesseladresse	
WAIT	– Warten auf E/A-Ende	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 6–12 Byte

f) Wirkung: Der Ruf bewirkt durch die Angabe von unterschiedlichen Geraetekommandos die ganze oder teilweise Loeschung des Nutzerbereiches der Bildschirmausgabebaugruppe. Dabei fuehren die einzelnen Geraetekommandos folgende Funktionen aus:

k = 1: Komplette Loeschung des Nutzerbereiches ohne Positions- und Laengenangaben

k = 2: Teilloeschung des Nutzerbereiches mit Angabe der Anfangs- und Endposition als Zeile und Spalte

k = 3: Teilloeschung des Nutzerbereiches mit Angabe der Anfangsposition als Zeile und Spalte und der Bereichslaenge.

g) Fehler:

Fehlercode=10H: Im Ruf sind fehlerhafte Parameter enthalten. Er wird nicht ausgefuehrt und die Task wird fortgesetzt.

3.6.4.3. Eingabe von Daten READ

3.6.4.3.1. READ fuer LBL 1210

a) Aufgabe: Dieser Ruf fuehrt die Datenuebertragung vom Lochbandleser 1210 zum internen Arbeitsspeicher durch.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] READ  R=n,(PTR=i,IN=k,BOB=(adr!symb),LOB=l
                 [,EOD=c][,ECA=(adr!symb)][,WAIT]
                 !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
PTR	- Lochbandleser	
i	- logische Geraetenummer	$0 < i < 256$
IN	- Geraetekommando	
k	- Nummer des Geraetekommandos	
BOB	- Anfangsadresse des Datenpufferspeichers	
LOB	- Einzulesende Daten	
l	- Anzahl der Bytes	$1 \leq l < 65536$
EOD	- Endezeichen der Daten	
c	- Definiertes hexadezimales Codezeichen	
ECA	- Fehlerschluesseladresse	
WAIT	- Warten auf E/A-Ende	
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 6-12 Byte

f) Wirkung: Der Ruf fuehrt eine Datenuebertragung vom Lochband-

leser zum internen Arbeitsspeicher unter wahlweiser Benutzung der Geraetekommandos fuer spezielle Funktionen durch.

Dabei haben die einzelnen Geraetekommandos folgende Funktionen:

k = 1: Vorwaertslesen ohne Beruecksichtigung von Steuer- und Funktionszeichen entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 2: Vorwaertslesen unter Beruecksichtigung von Steuer- und Funktionszeichen entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 3: Rueckwaertslesen ohne Beruecksichtigung von Steuer- und Funktionszeichen entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 4: Rueckwaertslesen unter Beruecksichtigung von Steuer- und Funktionszeichen entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 5: Vorwaertslesen ohne Beruecksichtigung von Steuer- und Funktionszeichen mit automatischer Paritaetskontrolle entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 6: Vorwaertslesen unter Beruecksichtigung von Steuer- und Funktionszeichen mit automatischer Paritaetskontrolle entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 7: Rueckwaertslesen ohne Beruecksichtigung von Steuer- und Funktionszeichen mit automatischer Paritaetskontrolle entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

k = 8: Rueckwaertslesen unter Beruecksichtigung von Steuer- und Funktionszeichen mit automatischer Paritaetskontrolle entsprechend der angegebenen Laenge der einzulesenden Daten oder bis zum Endezeichen.

g) Fehler:

Fehlercode=10H: siehe unter Pkt. 3.6.4.2.2, COTR fuer LBS 1215

Fehlercode=40H: Bandende oder Bandriss. Es wird der Ruf uebergangen und die Task fortgesetzt.

Fehlercode=80H: Paritaetsfehler auf Datentraeger. Dieser Fehler hat keine Auswirkung auf die Abarbeitung des Rufes.

3.6.4.3.2. READ fuer Folienspeicher MF 3200 READ

a) Aufgabe: Dieser Ruf fuehrt die Datenuebertragung vom Folienspeicher auf den Arbeitsspeicher durch.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] READ  R=n,(FD=i,BOB=(adr!symb),BPOS=t/r,AOR=1
                  [,WAIT][,ECA=(adr!symb)]
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
FD	- Folienspeicher	
i	- logische Geraetenummer	$0 < i < 256$
BOB	- Anfang des Pufferspeichers	
BPOS	- Anfangsangaben zur Position	
t	- Nummer der ersten Quellspur	$0 < t < 74$

r – Nummer des ersten Quellsektors $0 < r < 27$
AOR – Einzulesende Sektoren
l – Anzahl der einzulesenden Sektoren $0 < l < 257$
ECA – Fehlerschluesseladresse
WAIT – Warten auf E/A-Ende
PARA – Parameteradresse
adr – absolute Adresse $0 < adr < FFFFH$
symb – symbolische Adresse

d) Ruf-Nr: 8

e) Ruflaenge: 11-13 Byte

f) Wirkung: Die im Ruf notierte Anzahl von Sektoren mit einer konstanten Blocklaenge von 128 Datenbytes werden ab der Anfangsposition (Spur und Sektor) vom Folienspeicher in den Arbeitsspeicher uebertragen.

g) Fehler:

Fehlercode=10H: Im Ruf sind fehlerhafte Parameter enthalten. Der E/A-Ruf wird nicht ausgefuehrt.

Fehlercode=20H: Auf die aktuell zu bearbeitende Spur des Folienspeichers ist eine Positionierung nicht durchfuehrbar, da das Laufwerk defekt ist. Der E/A-Ruf wird abgebrochen.

Fehlercode=40H: Der zu bearbeitende Folienspeicher ist im Laufwerk nicht richtig positioniert und durch Bedienereingriff zu korrigieren.
Der E/A-Ruf wurde nicht eroeffnet.

Fehlercode=41H: Der sechste als defekt erkannte Sektor wurde in die Error-Datei eingelagert, wobei kein Datenverlust auftritt. Der Folienspeicher ist weiterhin verwendbar, sein Inhalt sollte jedoch umgehend kopiert werden. Der E/A-Ruf wird vollstaendig abgearbeitet.

Fehlercode=42H: Der siebente und achte als defekt erkannte Sektor wurde in die Error-Datei eingelagert,

wobei kein Datenverlust auftritt. Der Inhalt des Folienspeichers ist unbedingt zu kopieren. Der E/A-Ruf wird fortgesetzt.

Fehlercode=80H: Das Identifikationsfeld des aktuell zu bearbeitenden Datenfeldes ist nicht lesbar und der Ruf wird abgebrochen.
Die Diskette ist auf einem anderen Folienspeicher-Laufwerk zu lesen.

Fehlercode=81H: Das vom Folienspeicher gelesene und in den Arbeitsspeicher uebertragene Datenfeld weist einen CRC-Fehler bzw. einen Datenmarkenausfall auf. Der E/A-Ruf wird ordnungsgemaess beendet. Der Folienspeicher ist auf einem anderen Laufwerk alternativ abzuarbeiten.

Fehlercode=82H: Der neunte als defekt erkannte Sektor wurde auf dem Folienspeicher festgestellt, wobei keine Auslagerung in die Error-Datei moeglich ist. Gleichzeitig dazu erfolgt die Ausgabe der Systemnachricht SF07. Ein Datenverlust ist eingetreten, der Inhalt des Folienspeichers ist neu aufzubereiten. Der E/A-Ruf wird abgebrochen.

Fehlercode=82H: Die Aktualisierung der Error-Datei fuer die Auslagerung eines defekten Sektors ist fehlerhaft verlaufen und damit ein Datenverlust eingetreten. Der E/A-Ruf wird abgebrochen. Gleichzeitig dazu erfolgt die Ausgabe der Systemnachricht SF09.

Fehlercode=82H: Die Auslagerung eines defekten Sektors in die Error-Datei wurde erfolglos abgebrochen. Ein Datenverlust ist eingetreten, der E/A-Ruf wird abgebrochen. Gleichzeitig dazu erfolgt die Ausgabe der Systemnachricht SF10.

Fehlercode=83H: Der zu bearbeitende Sektor auf dem Folienspeicher ist nicht auffindbar.
Der Folienspeicher ist auf einem anderen Laufwerk zu bearbeiten.

3.6.4.3.3. READ fuer alphanumerische Tastatur READ

a) Aufgabe: Dieser Ruf uebertraegt zeichenweise die eingetasteten Informationen von der Tastatur K 7602 zum Arbeitsspeicher bzw. zur Bildschirmbaugruppe MON1.
b) Status: Ruf
c) Schreibweise:
Ruf:

```
[name] READ  R=n,(KEYB=i,IN=k[,BOB=(adr!symb)][,BPOS=s/z]
                  [,LOB=l][,MON=i][,WAIT]
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
KEYB	– Tastatur	
i	– logische Geraetenummer	$0 < i < 256$
IN	– Geraetekommando	
k	– Nummer des Geraetekommandos	
BOB	– Anfangsadresse des Datenpufferspeichers	
LOB	– Einzulesende Daten	
l	– Anzahl der einzutastenden Daten	$0 < l < 256$
BPOS	– Anfangsangaben zur Position	
s	– Spalte	$0 < s < 65$
z	– Zeile	$0 < z < 17$
MON	– Bildschirmausgabebaugruppe	
WAIT	– Warten auf E/A-Ende	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 8-13 Byte

f) Wirkung: Der Ruf fuehrt eine zeichenweise Datenuebertragung von der Tastatur zum internen Arbeitsspeicher bzw. zur Bildschirmbaugruppe unter wahlweiser Benutzung der Geraetekommandos durch.

Dabei haben die einzelnen Geraetekommandos folgende Funktionen.

k = 1: Uebertragung von zeichenweise eingetasteten Daten zur Bildschirmbaugruppe unter Angabe der Anfangsposition (Zeile und Spalte) sowie der Anzahl der einzugebenden Daten.

k = 2: Uebertragung von zeichenweise eingetasteten Daten zur Bildschirmbaugruppe unter Angabe der Anfangsposition (Zeile und Spalte) sowie der Anzahl der einzugebenden Daten mit anschliessender Uebertragung zum internen Arbeitsspeicher.

k = 3: Uebertragung eines beliebigen Tastaturzeichens in den internen Arbeitsspeicher.

3.6.4.4. Ausgabe von Daten

3.6.4.4.1. WRIT fuer LBS 1215 WRIT

a) Aufgabe: Dieser Ruf fuehrt die Datenuebertragung vom internen Arbeitsspeicher zum Lochbandstanzer 1215 durch.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] WRIT  R=n,(PTP=i,IN=k,BOB=(adr!symb),LOB=l
                 [,EOD=c][,ECA=(adr!symb)][,WAIT]
                 !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R   Registerrettung

n   - Registerrettungsart $0 \leq n \leq 2$
PTP - Lochbandstanzer
i   - logische Geraetenummer $0 < i < 256$
IN   - Geraetekommando
k   - Nummer der Geraetekommandos
BOB - Anfangsadresse des Datenpufferspeichers
LOB - Einzulesende Daten
l   - Anzahl der Bytes $1 \leq l < 65536$
EOD - Endezeichen der Daten
c   - Definiertes hexadezimales Codezeichen
ECA - Fehlerschluesseladresse
WAIT - Warten auf E/A-Ende
PARA - Parameteradresse
adr - absolute Adresse   0 < adr < FFFFH
symb - symbolische Adresse

d) Ruf-Nr: 8

e) Ruflaenge: 9-12 Byte

f) Wirkung: Der Ruf fuehrt eine Datenuebertragung vom internen Arbeitsspeicher zum Lochbandstanzer unter Beachtung der im Geraetekommando spezifizierten Kontrollfunktion durch.

Dabei haben die einzelnen Geraetekommandos folgende Funktionen:

k = 1: Stanzen mit automatischer Paritaetskontrolle bei Angabe der Laenge der auszustanzenden Daten oder bis zum Endezeichen.

k = 2: Stanzen ohne automatische Paritaetskontrolle bei Angabe der Laenge der auszustanzenden Daten oder bis zum Endezeichen.

g) Fehler:

Fehlercode=10H: siehe Pkt. 3.6.4.2.2, COTR fuer LBS 1215
Fehlercode=20H: Paritaets- oder Stanzfehler. Der Ruf wird abgebrochen und die Task fortgesetzt.
Fehlercode:40H: siehe Pkt. 3.6.4.2.2, COTR fuer LBS 1215
Fehlercode=80H: siehe Pkt. 3.6.4.2.2, COTR fuer LBS 1215

3.6.4.4.2. WRIT fuer SD 1156

a) Aufgabe: Dieser Ruf realisiert die Datenuebertragung vom internen Arbeitsspeicher zum Seriendrucker 1156.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] WRIT  R=n,(LP=i,IN=k,BOB=(adr!symb),LOB=l
                 [,BPOS=d][,FLF=(s!z),LEP=(1!2!3)]
                 [,ECA=(adr!symb)][,WAIT]
                 !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
LP	– Seriendrucker	
i	– logische Geraetenummer	$0 < i < 256$
IN	– Geraetekommando	
k	– Nummer des Geraetekommandos	
BOB	– Anfangsadresse des Datenpufferspeichers	
LOB	– Einzulesende Daten	
l	– Anzahl der Bytes	$0 < l < 65536$
BPOS	– Angaben zur Positionierung	
d	– Position der ersten Druckzeichen	$0 < d < 171$
FLF	– Angaben zur Formulartraegersteuerung	
s	– einmalige Seitenschaltung	
z	– Zeilenschaltung	$0 < z < 32$
LEP	– Formulartraegersteuerung	
1	– Leporello 1	
2	– Leporello 2	
3	– Leporello 1 und 2	
ECA	– Fehlerschluesseladresse	
WAIT	– Warten auf E/A-Ende	

PARA - Parameteradresse
adr - absolute Adresse   0 < adr < FFFFH
symb - symbolische Adresse

d) Ruf-Nr: 8

e) Ruflaenge: 10-14 Byte

f) Wirkung: Der Ruf fuehrt eine Datenuebertragung vom internen Arbeitsspeicher zum Seriendrucker 1156 unter Benutzung von im Geraetekommando spezifizierten Funktionen durch. Dabei haben die einzelnen Geraetekommandos folgende Funktionen:

k = 1: Aus dem Arbeitsspeicher erfolgt entsprechend der angegebenen Zeichenanzahl eine Ausgabe auf den Seriendrucker. Bei der wahlweisen Angabe einer Positioniergroesse wird die Positionierung vor dem Zeichendruck ausgefuehrt.

k = 2: Aus dem Arbeitsspeicher erfolgt entsprechend der angegebenen Zeichenanzahl eine Ausgabe auf den Seriendrucker. Bei der wahlweisen Angabe einer Positioniergroesse wird die Positionierung vor dem Zeichendruck ausgefuehrt. Nach beendetem Zeichendruck erfolgt ein Transport auf dem angegebenen Formularträger als Zeilen-oder Seitenvorschub fuer Leporello 1 und/oder 2.

g) Fehler:

Fehlercode=10H: siehe Pkt. 3.6.4.2.1, COTR fuer SD 1156

Fehlercode=20H: Durch einen Defekt am Geraet ist die Interruptfolge gestoert. Es wird der Ruf abgebrochen und die Task fortgesetzt.

Fehlercode=40H: siehe Pkt. 3.6.4.2.1, COTR fuer SD 1156

Fehlercode=80H: Fehlbedienung. Die Abarbeitung des Rufes wird nicht beeinflusst.

3.6.4.4.3. WRIT fuer Bildschirmbaugruppe MON1 WRIT

a) Aufgabe: Dieser Ruf fuehrt eine Datenuebertragung vom Arbeitsspeicher auf den Nutzerbereich der Bildschirmbaugruppe MON1 durch.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] WRIT  R=n,(MON=i,IN=k,BOB=(adr!symb),(EPOS=z/s!LOB=l)
                  [,BPOS=z/s][,ECA=(adr!symb)][,WAIT]
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
MON	– Bildschirmausgabebaugruppe	
i	– logische Geraetenummer	$0 < i < 256$
IN	– Geraetekommando	
k	– Nummer des Geraetekommandos	
BOB	– Anfangsadresse des Datenausgabepuffers	
BPOS	– Anfangsangaben zur Position	
EPOS	– Endangaben zur Position	
z	– Zeile	$0 < z < 17$
s	– Spalte	$0 < s < 65$
LOB	– Bereichslaenge	
l	– Anzahl der zu loeschenden Zeichen	$0 < l < 1025$
ECA	– Fehlerschluesseladresse	
WAIT	– Warten auf E/A-Ende	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 10-14 Byte

f) Wirkung: Der Ruf fuehrt eine Datenuebertragung vom internen Arbeitsspeicher auf die Bildschirmbaugruppe aus. Durch die Angabe von verschiedenen Geraetekommandos sind unterschiedliche Positionierungsvarianten auf der Bildschirmbaugruppe moeglich. Dabei haben die einzelnen Geraetekommandos folgende Funktionen:

k = 1: Die Datenausgabe aus dem Arbeitsspeicher erfolgt entsprechend der angegebenen Laenge ab Anfang des Nutzerbereiches.

k = 2: Die Datenausgabe aus dem Arbeitsspeicher erfolgt ab Anfang des Nutzerbereiches bis zur Endposition (Zeile und Spalte).

k = 3: Die Datenausgabe aus dem Arbeitsspeicher erfolgt entsprechend der angegebenen Laenge ab der Anfansposition (Zeile und Spalte).

k = 4: Die Datenausgabe aus dem Arbeitsspeicher erfolgt ab der angegebenen Anfangsposition (Zeile und Spalte) bis zur Endposition (Zeile und Spalte).

g) Fehler:

Fehlercode=10H: Im Ruf sind fehlerhafte Parameter enthalten. Es wird der Ruf nicht ausgefuehrt und die Task fortgesetzt.

3.6.4.4.4. WRIT fuer Folienspeicher MF 3200 WRIT

a) Aufgabe: Dieser Ruf fuehrt die Datenuebertragung von dem Arbeitsspeicher zum Folienspeicher durch.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] WRIT  R=n,(FD=i,BOB=(adr!symb),BPOS=t/r,AOR=l
                   [,WAIT][,ECA=(adr!symb)]
                   !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \le n \le 2$
FD	- Folienspeicher	
i	- logische Geraetenummer	$0 < i < 256$
BOB	- Anfang des Pufferspeichers	
BPOS	- Anfangsangaben zur Position	
t	- Nummer der ersten Zielspur	$0 < t < 74$
r	- Nummer des ersten Zielsektors	$0 < r < 27$
AOR	- auszugebende Sektoren	
l	- Anzahl der auszugebenden Sektoren	$0 < l < 257$
ECA	- Fehlerschluesseladresse	
WAIT	- Warten auf E/A-Ende	
PARA	- Parameteradresse	
adr	- absolute Adresse	$0 < adr < FFFFH$
symb	- symbolische Adresse	

d) Ruf-Nr: 8

e) Ruflaenge: 11-13 Byte

f) Wirkung: Die im Ruf notierte Anzahl von Sektoren mit einer konstanten Blocklaenge von 128 Datenbytes wird vom Arbeitsspeicher zum Folienspeicher ab der Anfangsposition (Spur und Sektor) uebertragen.

g) Fehler: siehe Pkt. 3.6.4.3.2, READ fuer Folienspeicher MF 3200

3.6.4.5. Geraetemanipulation

3.6.4.5.1. Geraeteumschaltung und -zuweisung ASGN

a) Aufgabe: Diese Funktion ermoeglicht die Veraenderung der Zuordnung von logischer Geraetenummer und physischem

Geraet zweier DV-Geraete. Eine Geraeteumschaltung setzt dabei stets ein defektes Geraet voraus, das durch ein Ausgabegeraet fuer Daten ersetzt wird. Die Geraetezuweisung wird dagegen fuer zwei DV-Geraete gleichen Typs durchgefuehrt, die bisher keine Daten ausgegeben haben und nicht defekt sind.

b) Status: Ruf oder Kommando

c) Schreibweise:

Kommando:

```
ASGN  SUD=i,SEQ=i
```

Ruf:

```
[name] ASGN  R=n,(SUD=i,SEQ=i[,ECA=(adr!symb)]
                 !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
SUD	- Defekt- oder erstes Tauschgeraet	
SEQ	- Ersatz- oder zweites Tauschgeraet	
i	- logische Geraetenummer	$0 < i < 256$
ECA	- Fehlerschluesseladresse	
PARA	- Parameteradresse	
adr	- absolute Adresse	$0 < adr <$ FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr: 9

e) Ruflaenge: 5-7 Bytes

f) Wirkung: Die Geraeteumschaltung oder-zuweisung erfolgt stets, wenn die notwendigen Voraussetzungen fuer deren Ausfuehrung gegeben sind. Bei nicht erfuellten Bedingungen werden diesbezueglich Systemnachrichten ausgegeben sowie ein Fehlercode bereitgestellt. Ein wiederholter Aufruf der Funktionen fuehrt zu einer

zwangsweisen Ausfuehrung, wobei evtl. auftretende Auswirkungen auf die weitere Betriebsweise vom Anwender beruecksichtigt werden muessen.

g) Fehler:

Fehlercode=40H: Eine Geraeteumschaltung oder -zuweisung ist nicht moeglich, da die Geraetetypen ungleich sind.

Fehlercode=40H: Eine Geraeteumschaltung oder -zuweisung ist nicht moeglich, da das angegebene Ersatzgeraet derzeitig arbeitet bzw. taetig gewesen ist. Die Ausgabe von verschiedenen Daten auf einen Datentraeger bei der weiteren Arbeit kann nicht ausgeschlossen werden.

3.6.4.5.2. Geraeterueckschaltung ASGN

a) Aufgabe: Die Geraeterueckschaltung ermoeglicht die Eingliederung von reparierten Geraeten unter der generierten logischen Geraetenummer.

b) Status: Kommando

c) Schreibweise:

Kommando:

ASGN SUD=i

Dabei bedeuten die Schluesselworte und Parameter:

SUD - Defektgeraet

i - logische Geraetenummer $0 < i < 256$

f) Wirkung: Das angegebene logische Geraet wird auf das bei der Generierung festgelegte physische Geraet zurueckgeschaltet. War das logische Geraet nicht gestoert, so wird die Geraeterueckschaltung uebergangen.

3.6.4.6. Test auf Ende eines E/A-Rufes WAIT

a) Aufgabe: Der Ruf testet den Bearbeitungszustand einer innerhalb der Task ausgeloesten E/A-Operation.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] WAIT R=n,(CADR=(adr!symb)!PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R - Registerrettung

n - Registerrettungsart $0 \leq n \leq 2$

CADR - Anfangsadresse des E/A-Rufes

PARA - Parameteradresse

adr - absolute Adresse 0 < adr < FFFFH

symb - symbolische Adresse

d) Ruf-Nr.: 10

e) Ruflaenge: 5 Byte

f) Wirkung: Anhand der im Ruf enthaltenen Adresse wird der aktuelle Bearbeitungszustand der E/A-Operation ueberprueft. Ist der unter der Adresse angegebene E/A-Ruf beendet worden, kann die Task weiter bearbeitet werden.

Wurde die E/A-Operation noch nicht beendet, so wird die laufende Task unterbrochen.

3.7. Organisation der Bedienerkommunikation

Die Kommunikation mit dem Echtzeitsteuerprogrammsystem erfolgt ueber Kommandos und Systemnachrichten. Die Arbeit mit diesen im EIEX 1521 standardmaessig implementierten Programmoduln setzen die alphanumerische interruptgesteuerte Tastatur K 7602 und die Bildschirmausgabebaugruppe MON1 K 7221 als geraetetechnische Basis voraus. Zusaetzlich koennen die Systemnachrichten auf dem Seriendrucker 1156 protokolliert werden.

3.7.1. Kommando-Organisation

3.7.1.1. Aufbau und Wirkung der Kommando-Organisation

Die Kommando-Organisation besteht aus folgenden Programmoduln:

- Pruefroutinen fuer eingegebene Parameter
- Steuerung der Ausgabe auf die Bildschirmbaugruppe MON1 und auf den Seriendrucker 1156
- Task zur Ausfuehrung des eingegebenen Kommandos

Jedes eingegebene Kommando wird auf seine Zulaessigkeit geprueft. Die benoetigten Parameter werden durch Ausgabe von relevanten Schluesselworten in Menue-Technik abgefordert und einer logischen Pruefung unterzogen.
Vor der Beendigung der Eingabe ist eine Korrektur der eingetasteten Parameter moeglich. Nach Abschluss der Eingabe eines Kommandos wird die eingegebene Zeile um die aktuelle Uhrzeit ergaenzt und in den geschuetzten Bereich fuer Systemnachrichten auf dem MON1 uebertragen, bzw. ueber den Seriendrucker 1156 ausgegeben.
Die Zeitdauer einer Kommando-Eingabe hat keinen Einfluss auf andere gleichzeitig verlaufende Prozesse.
Da fast alle EIEX-Kommandos einen adaequaten EIEX-Ruf besitzen, werden diese Kommandos in entsprechende Rufe transformiert und ueber den Start einer Task abgearbeitet.
Die Ausfuehrung eines Kommandos im EIEX 1521 sollte stets durch die Task1 erfolgen; es kann jedoch bei der Generierung eine beliebige Task festgelegt werden.
Der Start der Task zur Ausfuehrung des Kommandos erfolgt nur nach einer fehlerfreien Eingabe.
Waehrend der Abarbeitung eines Kommandos ist keine weitere Kommando-Eingabe moeglich.

Eigene, durch den Anwender selbst entwickelte Kommandos koennen in die Kommandoorganisation integriert werden. Die dazu notwendigen Pruefungs- und Rufroutinen sind durch den Anwender zu entwickeln und bei der Generierung in diese spezielle Echtzeitsteuerprogrammversion einzubinden.

3.7.1.2. Hinweise zur Benutzung der Kommando-Organisation

Durch Betaetigung der bei der Generierung definierten Gesuch-Taste wird die Kommando-Organisation aufgerufen. Die Bereitschaft zur Eingabe eines Kommandos wird durch Anzeige eines Sterns in der letzten Bildschirmzeile des Systemnachrichtenbereiches mit nachfolgend gesetztem Cursor bestaetigt.
Die erste Eingabe in diese Zeile ist stets das Kommando-Wort von max. 5 Alphazeichen, dessen Eingabe mit dem Betaetigen der generierten Start-Taste abgeschlossen wird. In Abhaengigkeit vom angewaehlten Kommando wird nachfolgend das erste Schluesselwort fuer die zugehoerige Parametereingabe ausgegeben (z.B. TASK=_) und der Parameterwert eingetastet. Ist die Eingabe von weiteren Parametern erforderlich, so muss nach jedem Parameterwert das bei der Generierung festgelegte Trennzeichen eingetastet werden. Die Reihenfolge der Schluesselworte, deren Anzahl sowie die entsprechende Wortlaenge werden durch die Kommando-Organisation gesteuert. Jedes Kommando wird mit dem Betaetigen der generierten Start-Taste beendet.
Bei der Aufforderung zur Eingabe von Zeitwerten (z.B. Kommando RUN) wird zur Bedienerunterstuetzung am Zeilenende der generierte Zeittakt des Echtzeitsteuerprogrammsystems angezeigt (z.B. CL = 100MS).
Vor dem Abschluss einer Parametereingabe sind folgende Korrekturen moeglich:

- Erneute Eingabe des Parameterwertes ab gewaehlter Cursorposition

- Loeschen des ueber dem Cursor stehenden Zeichens durch Druecken der DEL-Taste, wobei vorhandene nachfolgende Zeichen linksbuendig verschoben werden.
- Betaetigen der INS-Taste mit nachfolgendem Einfuegen von Zeichen vor der aktuellen Cursorposition, wobei nachfolgende Zeichen einschliesslich Cursor rechtsbuendig verschoben werden. Die INS-Funktion wird durch das Betaetigen einer anderen Funktionstaste oder beim Erreichen der Eingabekapazitaet beendet.

  Wurde die vorgeschriebene Stellenzahl im Eingabebereich bereits erreicht, so ist die INS-Funktion wirkungslos. Alle weiteren eingegebenen Zeichen werden wie bei der erneuten Parametereingabe behandelt.

Jedes eingetastete Kommando und die dazugehoerigen Parameterwerte werden geprueft. Bei einer fehlerhaften Eingabe erfolgt am Zeilenende eine spezielle Fehlerausschrift (z.B. COMMAND ERROR 5).
Durch Betaetigen der DEL-Taste wird der eingegebene Parameterwert automatisch geloescht und eine erneute Eingabe ist moeglich. Beim Druecken der Taste Cursor links (<---) wird die gesamte Kommando-Zeile geloescht und zur erneuten Kommando-Eingabe aufgefordert.
Jedes fehlerfreie Kommando wird am Zeilenende mit der aktuellen Uhrzeit versehen in den Systemnachrichtenbereich des MON1 umgespeichert.
Der Ausgabebereich fuer Systemnachrichten auf dem MON1 betraegt standardmaessig 4 Zeilen. Seine Groesse kann bei der Generierung im Bereich von 1 bis 12 Zeilen festgelegt werden.
Duch das Kommando LOG koennen alle eingegebenen Kommandos sowie die Systemnachrichten zusaetzlich auf dem Seriendrucker SD 1156 protokolliert werden. Das Kommando NOLOG beendet die Druckprotokollierung.

3.7.1.3. EIEX-Kommandos

Die Kommando-Organisation gestattet die Eingabe und Ausfuehrung folgender Kommandos, die standardmaessig in EIEX 1521 enthalten sind:

ASGN - Geraetezuweisung, Geraeteum- und -rueckschaltung
CNCL - Abbrechen der Taskbearbeitung
RUN - Starten der Task
DISP - Verhindern der Taskbearbeitung
ENAP - Erlauben der Taskbearbeitung
GO - Fortsetzen einer Task
SDATE - Eingabe des Datums
STIME - Eingabe der Uhrzeit
LOG - Protokollierung von Systemnachrichten ueber Seriendrukker
NOLOG - Beenden der Protokollierung von Systemnachrichten ueber Seriendrucker
HELP - Wechsel der Prioritaet einer Task
CHAN - Wechsel der Prioritaet zweier Tasks
OPEN - Eroeffnen einer Datei von Folienspeicher
CLOS - Schliessen einer Datei auf Folienspeicher
POSF - Positionieren einer sequentiellen Datei auf Anfang
FIDE - Vereinbaren einer Datei auf Folienspeicher
CFID - Loeschen einer Datei auf Folienspeicher
INIT - Schreiben des Datentraegerkennsatzes

In der Anlage 2, Uebersicht der EIEX-Kommandos, sind alle standardmaessig in EIEX 1521 enthaltenen Kommandos und deren Parameter dargestellt.

3.7.2. Systemnachrichten-Organisation

3.7.2.1. Aufbau und Wirkung der Systemnachrichten-Organisation

Die Systemnachrichten-Organisation besteht aus folgenden Programmoduln:

- Generierbarer Ausgabepufferbereich
- Pufferverwaltung
- Generierbare Aufbereitung der Systemnachrichten
- Ausgabe von Systemnachrichten ueber eine generierte Task

Die Systemnachrichten beinhalten Hinweise zum Arbeitsstand des Echtzeitsteuerprogramms EIEX 1521 sowie Fehlermeldungen. Die Uebergabe der Systeminformation an den Ausgabepuffer erfolgt ueber die Pufferverwaltung, die auch das Ausgabeprogramm als Task anmeldet.
Das Ausgabeprogramm sollte stets als Task 2 fuer den MON1 und den Seriendrucker 1156 in EIEX 1521 eingebunden werden, es kann jedoch eine andere Prioritaet bei der Generierung festgelegt werden. Fuer anwendungsspezifische Loesungen kann ein Programmmodul von EIEX 1521 zur Aufbereitung einer Ausgabezeile genutzt werden, so dass prinzipiell die Ausgabe von Systemnachrichten auf andere periphere Geraete moeglich ist.
Eine Sytemnachricht hat folgenden standardisierten symbolischen Aufbau:

>HR:MI:SE< Sxy z adr TEXT:R

Die verwendeten Abkuerzungen der Systemnachricht haben folgende Bedeutung:

HR - Uhrzeit in Stunden
MI - Uhrzeit in Minuten
SE - Uhrzeit in Sekunden
S - Systemnachricht
x - Kurzzeichen der meldenden Systemkomponente
- B BAB-Treiber
- C geraetespezifische Zeitueberwachung
- D DV-Rahmensteuerung
- E Systemnachrichten-Organisation
- F Folienspeicher-Treiber
- G Status-Fehler der DV-Geraete
- H File-Haendler
- I Interrupt-Organisation
- K Tastatur-Treiber
- L LBL-Treiber
- P Prozess-Rahmensteuerung und Treiber
- R EIEX-Rufe

s LBS-Treiber
t Echtzeit-Organisation
v Vorrang-Organisation
w SD-Treiber

y - laufende Fehlernummer der Systemkomponente
z - Fehlerverursachende Tasknummer
adr - Adresse des fehlerausloesenden Rufes
TEXT - Mnemotechnische Ausschrift
R - Kennzeichen fuer notwendige Bedienerreaktion

Wird eine Systemnachricht durch interne Kontrollen des Echtzeitsteuerprogrammsystems ausgeloest, werden den Symbolen z der Buchstabe S und adr der Wert Null zugeordnet. Entfaellt eine Bedienerreaktion, so ist das Kennzeichen R nicht in der Systemnachricht enthalten. Der Ausgabepufferbereich ist in seiner Groesse generierbar (Anzahl der Eintragungen mal 10 Byte) und wird dynamisch verwaltet. Werden waehrend der laufenden Ausgabe mehr Systemnachrichten in den Ausgabepuffer eingetragen, als die generierte Anzahl der Eintragungen zulaesst, so laeuft der Puffer ueber, wobei ein Verlust von Ausschriften eintritt.

3.7.2.2. Hinweise zur Reaktion

Systemnachrichten informieren stets den Bediener. Sie koennen aber auch Entscheidungen fordern, um eine stabile Weiterarbeit im Echtzeitbetrieb zu sichern.
Ist in einer Systemnachricht das Reaktionskennzeichen enthalten, so muss immer eine Bedienerhandlung erfolgen. Als Reaktion kann die Eingabe eines Kommandos oder das Herstellen der Arbeitsbereitschaft eines peripheren Geraetes erforderlich sein.
Die Systemnachrichten sind unter Beachtung der Hinweise aus Anlage 3, Systemnachrichten und Fehlerausschriften EIEX 1521, gruendlich auszuwerten, da durch die getroffenen Entscheidungen die Weiterarbeit im Echtzeitbetrieb massgeblich beeinflusst wird.

3.8. Die Ein/Ausgabe- Organisation fuer die Prozessperipherie

Die Programmodule der Ein/Ausgabe- Organisation fuer die Gereate der Prozessperipherie sind generierbarer Bestandteil des EIEX 1521. Dabei wird der Datenaustausch zwischen dem internen Speicher (Puffer) und den Prozess-Steckkarten des Systems ursadat 5000 organisiert.
Die Datenuebertragung wird von der im Applikationsprogramm stehenden Task angefordert. Die Bedienung der angesprochenen Prozess-Steckkarte(n) erfolgt entweder im ungeteilten oder im geteilten Verkehr. Im geteilten Verkehr bleibt das logische Geraet, das in der Task zum Datenverkehr aufgefordert wurde, fuer weitere E/A-Anforderungen bis zur Beendigung des eingeleiteten Verkehrs gesperrt. Der geteilte Verkehr laeuft immer interruptgesteuert ab. Beim geteilten Verkehr organisiert EIEX 1521 wahlweise einen Wartebetrieb.

3.8.1. Verkehr zwischen Applikationsprogramm und Prozessperipherie

Zur Anforderung des Datenverkehrs mit der Prozessperipherie des Systems ursadat 5000 stehen Rufe fuer folgende logische Geraetetypen zur Verfuegung:

- DAS Digitale Ausgabe Statisch
- DAD Digitale Ausgabe Dynamisch
- DES Digitale Eingabe Statisch
- DESM Digitale Eingabe Statisch ueber Multiplexer
- DESU Digitale Eingabe Statisch mit PU-Signal
- DED Digitale Eingabe Dynamisch
- DEAS Digitale Ein/Ausgabe Statisch

- IA Impulsausgabe
- UIZ Universeller Impulszaehler
- AEG Analog-Eingabe ueber die Prozess-Steckkarte AE-G
- AA1K Analog-Ausgabe ueber die Prozess-Steckkarte AA-1K
- AA5K Analog-Ausgabe ueber die Prozess-Steckkarte AA-5K
- ZI serielles Zwischenblock-Interface (zur Zeit noch in Entwicklung)
- UEW Zentrale Busueberwachung (in Entwicklung)

Die Rufe leiten den Verkehr mit der Prozessperipherie ueber Systemkomponenten des EIEX 1521 ein, die bei der Systemgenerierung entsprechend dem Einsatzfall aktiviert werden. Dabei erfolgt die Zuordnung einer Systemleistung (Ruf) zu den Systemkomponenten (Geraete-Treiber) ueber die im Ruf anzugebende logische Geraete-Nr. im Bereich 0-127. Jedem logischen Geraet ist eine Zuweisungstabelle zugeordnet. Die Verbindung zwischen logischem Geraet und der Zuweisungstabelle wird ueber eine Geraete-Adresstabelle realisiert, in welcher bei der Systemgenerierung die Adresszeiger der Zuweisungstabellen eingetragen werden. Dabei ergibt sich die logische Geraete-Nr. des logischen Geraetes aus der Position dieses Adresszeigers in der Geraete-Adresstabelle. In jeder Zuweisungstabelle sind folgende fuer die Arbeitsweise des logischen Geraetes relevanten Informationen enthalten, die entsprechend dem Einsatzfall von EIEX 1521 zur Systemgenerierung bereitgestellt werden muessen:

- Adresse der zugeordneten E/A-Tabelle im RAM
- Adresse der Initialisierungsroutine
- Adresse der Treiberroutine
- Geraetespezifische Parameter

Die Geraeteadresstabelle, die Zuweisungstabellen sowie die benoetigten Speicherbereiche fuer die E/A-Tabellen werden bei der

Systemgenerierung in EIEX 1521 eingebunden und sind dann nicht mehr zu veraendern.
Die Eigenschaften eines logischen Geraetes sind durch den Typ der Prozess-Steckkarte(n), deren Initialisierung, der projektabhaengigen Zuordnung der Systemkomponenten (Treiber- und Interruptservice-Routinen) und im definierten Mass von den Parameter-Options in den Rufen bestimmt.

3.8.2. Organisation des Verkehrs mit der Prozessperipherie

Die Initialisierung der Prozess-Steckkarten erfolgt beim Systemanlauf anhand der generierten Geraete-Adresstabellen.
Dabei wird ueber die Zeiger zur Zuweisungstabelle zu den einzelnen Initialisierungsroutinen verzweigt, deren Adressen in den Zuweisungstabellen enthalten sind. In EIEX 1521 stehen fuer jeden Typ der generierten E/A- Treiber Systemprogramme fuer die Initialisierung zur Verfuegung, die nach der noetigen Parameterversorgung als Unterprogramme in den Initialisierungsroutinen aufgerufen werden koennen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess- Geraetetreiber). Falls keine Parameterversorgung erforderlich ist, sind diese Initialisierungsroutinen Bestandteil der Systemkomponenten. Anderenfalls gehoeren sie zum Applikationsprogrammsystem (siehe Initialisierung der einzelnen Geraetetypen).
Der Verkehr mit den Geraeten der Prozessperipherie wird durch einen entsprechenden Ruf eroeffnet und fuehrt zum Eintritt in die Prozess-Rahmensteuerung.
Aus den im Ruf angegebenen Parametern (Rufparameter) wird die logische Geraetenummer gelesen und auf Zulaessigkeit geprueft. Fuer ein definiertes logisches Geraet werden die in der zugeordneten Zuweisungstabelle enthaltenen Informationen ausgewertet und in die angegebene Treiberroutine verzweigt. Die angesprochene Treiberroutine eroeffnet den E/A-Verkehr, wobei sie ihn bei ungeteiltem Verkehr vollstaendig durchfuehrt.
Ist im geteilten Verkehr das angesprochene logische Geraet besetzt, so wird die Task, die die E/A-Anforderung ausloesen moechte, in einem Wartepuffer mit einem einzigen Element, ggf.

unter Ausgabe einer Systemwarnung, gespeichert. Sind mehrere E/A-Anforderungen fuer ein besetztes logisches Geraet vorhanden, so wird die Task mit der hoechsten Prioritaet in den Wartepuffer eingetragen, waehrend die Bearbeitung der uebrigen Task, ggf. unter Ausgabe einer Systemwarnung, in ihrer Bearbeitung zurueckgestellt werden.
Die Versorgung der Treiberroutine mit Parametern erfolgt aus den geraetespezifischen Parametern, der Zuweisungstabelle und den Rufparametern. Beim interruptgesteuerten E/A-Verkehr wird die zugehoerige E/A-Tabelle durch den Treiber mit den Informationen gefuellt, die fuer die Durchfuehrung des Interruptbetriebes notwendig sind.
Die Bedienung der Interruptanforderung der Prozessperipherie uebernimmt eine dem Treiber zugeordnete Interrupt-Service-Routine (ISR). Der Eintritt in diese ISR ist im Applikationsprogramm abzusichern. Dazu muss in der Interruptvektortabelle ein Adresszeiger auf eine Befehlsfolge enthalten sein, in der die generierte Registerrettungsroutine (EI.ISR) aufgerufen und die von der ISR geforderten Parameter bereitstellt werden. Wenn nicht anders angegeben, ist in das HL-Register die Adresse der E/A- Tabelle des interruptausloesenden logischen Geraetes zu laden und zur System-ISR zu verzweigen.
Es ist dabei zu beachten, dass die Vektortabelle bei der Systemgenerierung definiert werden muss und damit ein unveraenderlicher Bestandteil der generierten Variante von EIEX 1521 darstellt.
Der Austritt aus der ISR in das System EIEX 1521 garantiert, dass diejenige Task weiter bearbeitet wird, die zum Zeitpunkt des Austritts die hoechste Prioritaet besitzt.
Die Beschreibung der Wirkungen der in EIEX 1521 bereitgestellten Initialisierungsroutinen, Treiber- und Interruptserviceroutinen sind den nachstehenden Beschreibungen der logischen Geraetetypen zu entnehmen.
Rufaufloesungen, Anwendungshinweise, der Aufbau der Zuweisungs- und E/A-Tabellen und eine Zusammenstellung der Fehlernachrichten sind den Anlagen 3 (Fehlerauschriften und Systemnachrichten EIEX 1521), 6 (Anwendungsmoeglichkeiten der Prozess- Gereate-

treiber), 7 (Rufbeschreibung der Prozess- Rufe) und 8 (Zuweisungs- und E/A- Tabellen der Prozess- Geraetetreiber) zu entnehmen.

3.8.3. Benutzerhinweise

Wenn bei der Systemgenerierung die Verarbeitung indirekter Rufparameter zugelassen wird, kann die Notation der Parameter direkt im Ruf oder in einem gesonderten Rufparameterblock erfolgen, auf den ein Adresszeiger verweist, der mit dem Schluesselwort PARA definiert wird. Stets ist dabei darauf zu achten, dass die im Ruf enthaltenen Parameter nur von den dazugehoerigen Treiberroutinen entsprechend des Kartentyps ordnungsgemaess erkannt und verarbeitet werden.
Die Rufparameter dienen der Veraenderung der generierten Eigenschaften des logischen Geraetes in einem durch die Treiberroutine abgesteckten Rahmen. Die zulaessigen Parameter und deren Wirkungen sind aus den Beschreibungen der logischen Geraetetypen zu entnehmen.
Werden Rufe bei der Arbeit im geteilten Verkehr angewendet, so wird die Angabe des Schluesselwortes WAIT empfohlen. Damit uebernimmt EIEX 1521 die Synchronisation zum Applikationsprogramm bei der Abarbeitung des Rufes eigenstaendig und gewaehrleistet eine optimale und fehlerfreie Arbeitsweise des logischen Geraetes. Wird die WAIT-Funktion nicht im Ruf notiert, dann muss der Anwender selbst fuer die zeitliche Synchronisation sorgen und kann sich des Rufes WAIT als generierbare Komponente von EIEX 1521 bedienen.

3.8.4. EIEX-Rufe

3.8.4.1. Digitale Ausgabe Statisch DAS

a) Aufgabe: Sofortige Ausgabe von digitalen Signalen im ungeteilten Verkehr aus einem Ausgabepuffer in einer der folgenden, fuer das angesprochene logische Geraet

bei der Systemgenerierung festgelegten Arten:

- Ausgabe eines Bytes ueber ein 8-Bit-Port, dessen physische Adresse in der Zuweisungstabelle angegeben ist (DASE).
- Ausgabe eines Bytes mit Geraetekontrolle ueber ein 8-Bit-Port, dessen physische Adresse in der Zuweisungstabelle angegeben ist (DASH, DASKT).
- Ausgabe von 2 Bytes ueber 8-Bit-Port, dessen physische Adresse in der Zuweisungstabelle angegeben ist (DASDI).
- Ausgabe eines Blockes von max. 255 Bytes ueber die Ports, deren physische Adressen in einem Portverzeichnis angegeben sind (DASBI). Dabei ist die Adresse des Portverzeichnisses in der Zuweisungstabelle definiert.

b) Status: Ruf

c) Schreibweise:

```
[name] POUT  R=n, (DAS=i [,ECA=(adr!symb)]
                         [,BOB=(adr!symb)]
                         [,(MASK=m!POM=(adr!symb))]
                       ! PARA = (adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
DAS	–	digitale Ausgabe statisch	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adresse Fehlerbyte	
BOB	–	Adresse Ausgabepufferbereich	
MASK	–	direktes Ausgabe-Maskenbyte	
POM	–	Zeiger auf Ausgabemasken-Bereich	
PARA	–	Adresse indirekter Rufparameterblock	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	
m	–	Maskenbereich	$0 < m < 256$

d) Ruf-Nr.: 7

e) Ruflaenge: 4-9 Byte

f) Wirkung: Fuer alle Systemkomponenten des logischen Geraetetyps DAS werden bei Angabe einer Fehlerbyte-Adresse im Ruf unabhaengig von einer evt. Erzeugung einer Systemnachricht Fehlercodes in den adressierten Speicher uebertragen.
Die moeglichen Fehlercodes fuer alle Geraete sind: 0, 42H, 43H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).
Fuer DASH und DASKT werden fehlerhafte Kanaele durch "1" in der zugeordneten Bitposition gekennzeichnet.

DASE Ausgabe eines Bytes:

Von einem in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer wird ein Byte ueber das in der Zuweisungs-Tabelle adressierte Port ausgegeben.
Eine Maskendefinition wird ignoriert.

DASH Ausgabe eines Bytes ueber die Prozesskarte DAS-H:

DASH gibt ein Byte von einem in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer ueber das in der Zuweisungs-Tabelle definierte Port aus. Nach dem Abklingen der Prellvorgaenge an den Relais-Ausgaengen (10 ms) wird durch Ruecklesen der Parallel-Kontakte der Haftrelais die ordnungsgemaesse Ausgabe ueberwacht. Bei Aktivierung der WAIT-Funktion verharrt die rufende Task in dieser Zeit im Ruf, waehrend das uebrige System weiterarbeitet.

DASKT Ausgabe eines Bytes ueber die Prozesskarte DAS-KT:

Der Treiber DASKT gibt ein Byte von einem in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer ueber Port 0 der in der Zuweisungs-Tabelle zugeordneten Prozesskarte DAS-KT aus.

Dabei wird durch Ruecklesen die ordnungsgemaesse Arbeit der Prozesskarte ueberwacht.

DASDI Ausgabe eines Doppel-Bytes:

Von einem in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer werden 2 Byte ueber 2 Ports ausgegeben, wobei das 1. Port in der Zuweisungs-Tabelle adressiert wird.

* Variante DASD1:

Die Adresse des 2. Port ist die um 1 vergroesserte Adresse des 1. Port

* Variante DASD2:

Die Adresse des 2. Port ist die um 2 vergoesserte Adresse des 1. Port

DASBI Ausgabe eines Datenblocks

Von einem in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer werden ueber max. 255 Port, deren Adressen in einem in der Zuweisungs-Tabelle adressierten Port-Verzeichnis enthalten sind, Datenbytes ausgegeben.
Ist eine Ausgabe-Maske definiert, so werden die Ports nicht bedient, deren Adressen in der entsprechenden Bitposition der Maske mit "1" maskiert sind. Dabei ist Bit 0 des ersten Masken-Bytes der 1. Adresse im Portverzeichnis fest zugeordnet.
Bei der maskierten Ausgabe ist zu beachten, dass die Ausgabe aus dem Pufferbereich stets lueckenlos erfolgt.

* Variante DASB1:

Die Ausgabe-Maske kann nur eine Laenge von einem Byte besitzen. Werden Bloecke mit einer Laenge > 8 Byte maskiert ausgegeben, so werden die restlichen Ports unmaskiert bedient.

g) Standards: Wird die Adresse eines Ausgabepuffers im Ruf nicht angegeben, dann gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse. Wird im Ruf keine Maske definiert, so erfolgt die Ausgabe unmaskiert.

h) Initialisierung: Fuer die Treiber vom Typ DAS stehen die entsprechenden Standard -Initialisierungs-Routinen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) zur Verfuegung. Der Anwender kann in der Initialisierungs-Routine fuer das betreffende logische Geraet, deren Adresse in der Zuweisungs-Tabelle definiert ist, die geeignete Routine als Unterprogramm aufrufen, nachdem die Parameterversorgung fuer diese Routine sichergestellt worden ist.
Fuer den Typ DAS werden die PIO's des logischen Geraetes (soweit vorhanden) auf "Byte-Ausgabe" initialisiert, die Zeitgeber (soweit vorhanden) werden gestoppt.

Initialisierung DASE, DASKT, DASDI:

Die Adressen der zugeordneten Initialisierungs-Routinen koennen direkt in die Zuweisungs-Tabelle eingetragen werden, da sich diese Routinen mit den noetigen Parametern selbst versorgen.

Initialisierung DASH:

In der System-Initialisierungsroutine wird die aktuelle Stellung der Haftrelais in die jeweils zugeordnete E/A-Tabelle uebertragen und die Adresse des Eintritts in eine System-Zeitroutine in die Zeitueberwachungstabelle eingetragen. Die System-Zeitroutine ist eine ISR, die von der Zeitorganisation angerufen werden kann. Im Eintritt in diese Routine ist in HL die Adresse der zugehoerigen E/A-Tabelle zu laden und zur System-Zeitroutine zu verzweigen.
Parameter:

< BC > = Adresse des Eintritts in die System-Zeitroutine

Initialisierung DASBI:

Die Initialisierung erfolgt anhand des in der Zuweisungs-Tabelle adressierten Port-Verzeichnisses. Dabei sind durch eine Maske "INIT-PIO" diejenigen Port-Adressen zu maskieren, die nicht auf einem PIO arbeiten. Dazu wird die entsprechende Bitposition mit "1" belegt, wobei Bit 0 des ersten Masken-Bytes der ersten Portadresse zugeordnet ist.
Weiterhin ist eine Definition einer Maske "INIT-CTC" erforderlich. Diese ist zweckmaessig so zu gestalten, dass fuer jede Prozesskarte ein Port aus dem Portverzeichnis unmaskiert bleibt. Diese Massnahme bewirkt, dass der als Zeitgeber fungierende CTC-Kanal 0 jeder DAS-Prozesskarte gestoppt wird.
Parameter:

< DE > = Adresse der Maske "INIT-PIO"

< HL > = Adresse der Maske "INIT-CTC"

3.8.4.2. Digitale Ausgabe Dynamisch DAD

a) Aufgabe: Ausgabe von digitalen Signalen mit einer programmierbaren zeitlichen Begrenzung im Bereich 0.1ms - 6.6s im geteilten Verkehr in folgenden, fuer das angesprochene logische Geraet bei der Generierung

festgelegten Arten:

- Ausgabe von 1-255 Saetzen mit einer Laenge von einem Byte ueber ein 8-Bit-Port, dessen physische Adresse in der Zuweisungstabelle angegeben ist (DADE).
- Ausgabe von 1-255 Saetzen mit einer Laenge von 2 Byte ueber ein 2 x 8-Bit-Port, dessen physische Adresse in der Zuweisungs-Tabelle angegeben ist (DADDI).
- Ausgabe von 1-255 Saetzen mit einer maximalen Laenge von 255 Bytes ueber die Ports, deren physische Adresse in einem Portverzeichnis angegeben sind (DADBI).

  Die Adresse des Portverzeichnisses wird in der Zuweisungs-Tabelle angegeben.

b) Status: Ruf

c) Schreibweise:

```
[name] POUT  R=n,(DAD=i[,ECA=(adr!symb)]
                       [,BOB=(adr!symb)]
                       [,(MASK=m!POM=(adr!symb))]
                       [,REC=r]
                       [,MASTER=s]
                       [,BOT=(adr!symb)]
                       [,WAIT]
                                            ! PARA=(adr!symb))
```

Dabei bedeuten die Schluesselworte und Parameter:

R	-	Registerrettung	
n	-	Registerrettungsart	$0 \leq n \leq 2$
DAD	-	digitale Ausgabe dynamisch	
i	-	logische Geraetenummer	$0 < i < 256$
ECA	-	Adresse Fehlerbyte	
BOB	-	Adresse Ausgabepufferbereich	
MASK	-	direktes Ausgabemasken-Bte	
POM	-	Zeiger auf Ausgabemaske	

REC - Anzahl der Datensaetze 0 < r < 256
MASTER - steuerndes (interruptsendendes) Port
s - Portnummer 0 < s < 256
BOT - Adresse eines Bereiches mit neuen Impulszeitkonstanten
WAIT - Aktivierung der WAIT-Funktion
PARA - Adresse des indirekten Parameterblocks
adr - absolute Adresse 0 < adr < FFFFH
symb - symbolische Adresse

d) Ruf-Nummer: 7
e) Ruflaenge: 4-13 Bytes
f) Wirkung:

Die Treiber vom Typ DAD haben die Aufgabe den Datentransfer einzuleiten, indem sie den ersten Datenblock vom Puffer auf die Ports des logischen Geraetes ausgeben (die Signale werden byteorientiert nach Ablauf der Impulszeit auf "LOW" geschaltet. Ferner werden alle Angaben, die sich auf den aktuellen Stand des Transfers beziehen, in der zum logischen Geraet zugehoerigen E/A-Tabelle zwischengespeichert.

Diese Informationen werden von den zugeordneten Interrupt-Service-Routinen zur Fortfuehrung bzw. Beendigung der Datenausgabe benutzt. Die Adressen zum Eintritt in diese IS-Routinen sind bei der Generierung entsprechend der Initialisierung in die Interrupt-Vektor-Tabelle einzubringen.

In diesem projektabhaengigen ISR-Eintritt ist die generierte Register-Rettungsroutine (EI.ISR) zu rufen, die Adresse der E/A-Tabelle des betreffenden logischen Geraetes nach HL zu laden und zur System-ISR zu verzweigen.

Fuer alle Systemkomponenten des logischen Geraetetypes DAD werden bei Angabe einer Fehlerbyte-Adresse im Ruf unabhaengig von einer evt. Erzeugung einer Systemnachricht Fehlercodes in den adressierten Speicher uebertragen. Die moeglichen Fehlercodes sind:

0, 42H, 43H, 45H, 46H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

In allen Rufen des logischen Geraetetypes DAD ist die Aenderung der Laenge der Impulse byteweise moeglich, wenn im Ruf die Adresse einer "Liste von Impulszeiten" angegeben wird. Die Liste der Impulszeiten muss folgenden Aufbau besitzen:

Byte-Nr.	Bedeutung
1	n = Anzahl der Port-Nummern
2	1. Port-Nummer
3	Zeitkonstante fuer 1. Port-Nr.
. . .	
m	n.-te Port-Nummer
m+1	Zeitkonstante fuer n.-te Port-Nr.

Sind die Portadressen in einem Portverzeichnis enthalten, so ist die Port-Nummer mit dem Index in diesem Verzeichnis identisch, sonst gilt die angegebene (1.) Port-Adresse als Port-Nr.1. Die Laufzeit des Zeitgebers ermittelt sich zu:

$$T_i = t_s * 256 * T_t * T_z$$

wobei

t_s die Dauer des Systemtaktes

T_t die Zeitkonstante des Grundtaktgebers

T_z die Zeitkonstante des Zeitgebers

bedeuten.

Es ist dabei zu beachten, dass die Zeitkonstante des Grundtaktgebers bei der Initialisierung auf die betreffenden Prozesskarten uebertragen wird und durch Rufe nicht mehr veraendert werden kann. Weiterhin ist zu beachten, dass nicht unbe-

dingt jedes Port einen eigenen Zeitgeber besitzen muss (siehe Geraetebeschreibung des Systems ursadat 5000).
Der Master ist derjenige Port, der fuer das logische Geraet eine Interruptanforderung senden kann. Diese Interruptanforderung wird mit dem Ende der Signale, die ueber die Portkanaele des Masters ausgegeben wurden, vom zugehoerigen Zeitgeber erzeugt.
Bei der Angabe eines neuen Masters im Ruf wird die Interruptanmeldung durch den urspruenglichen Master blockiert, waehrend sie fuer den neuen Master freigegeben wird.
Die Angabe des Masters erfolgt in Form der Port-Nr..
Wird im Ruf eine Anzahl von Datensaetzen definiert, so wird durch die generierte Treiber-Routine zunaechst der erste Satz ausgegeben. Mit Anerkennung des vom Master angemeldeten Interrupts wird ueber die anzuspringende ISR ein weiterer Datensatz entsprechend der im Ruf angegebenen Satzanzahl an das logische Geraet ausgegeben.

DADE Ausgabe von Bloecken zu einem Byte

Von einem in der Zuweisungs-Tabelle oder im Ruf definierten Puffer werden bis zu 255 Bloecke zu je einem Byte ueber den in der Zuweisungs-Tabelle angegebenen Port ausgegeben, wobei alle Signale nach dem Ablauf der Impulszeit auf "LOW" geschaltet werden. Fuer die dynamische Ausgabe von Einzelbytes muss das dem logischen Geraet zugeordnete Port interruptfaehig sein.
Eine Maskendefinition sowie die Angabe eines Masters werden ignoriert.
Bei der Angabe von neuen Impulszeiten ist nur die Angabe der Port-Nr. 1 zulaessig.

DADDI Ausgabe von Bloecken zu 2 Byte

Von einem in der Zuweisungstabelle oder im Ruf definierten Puffer werden bis zu 255 Bloecken zu je 2 Byte ueber zwei

Ports ausgegeben, von denen der erste Port in der Zuweisungs-Tabelle angegeben ist.
Eine Maskendefinition wird ignoriert.

* Variante DADD1

Die Adresse des 2. Port ist die um 1 vergroesserte Adresse des 1. Port. Der Master kann einer der beiden Ports sein, die initialisierte Zuordnung kann ueber Ruf variiert werden.

* Variante DADD2

Die Adresse des 2. Port ist die um 2 vergroesserte Adresse des 1. Port. Als Master kann nur der erste Port benutzt werden. Diese Zuordnung kann nicht veraendert werden; entsprechende Rufparameter werden ignoriert.
Bei der Angabe neuer Impulszeiten ist nur die Angabe der Port-Nr. 1 zulaessig.

DADBI Ausgabe von Bloecken bis zu max. 255 Byte

Von einem in der Zuweisungs-Tabelle oder im Ruf definierten Puffer werden ueber max. 255 Port bis zu 255 Bloecken mit max. 255 Byte ausgegeben. Die Bedienung der Ports erfolgt in der Reihenfolge der Notation der physischen Adressen in dem Port-Verzeichnis, das in der Zuweisungs-Tabelle des logischen Geraetes adressiert ist. Wird eine Ausgabemaske definiert, so werden die Ports nicht angesprochen, deren Adresse in der entsprechenden Stelle der Ausgabemaske mit "1" maskiert sind.
Dabei ist Bit 0 des ersten Maskenbytes der ersten Portadresse fest zugeordnet.
Die Angabe erfolgt dabei in jedem Fall aus einem lueckenlosen Pufferbereich heraus.
Die initialisierte Adresse des Masters kann ueber Ruf auf eine andere interruptfaehige Portadresse verlegt werden. Die

im Ruf anzugebende Port-Nr. entspricht dabei der Position dieser Portadresse im Portverzeichnis.
Im Ruf kann gleichzeitig eine beliebige Auswahl der im Portverzeichnis notierten Ports unter Angabe der Port-Nr. mit einer Impulszeit belegt werden, indem im Ruf eine Liste von Impulszeiten adressiert wird.
Es ist zu beachten,dass bei einigen Prozesskarten 2 Ports mit einem Impulsgeber verbunden sein koennen, so dass ungewollte Beeinflussungen der Impulszeit moeglich sind (siehe Technische Beschreibung der Prozesskarten des Systems ursadat 5000).
Ferner ist unbedingt darauf zu achten, dass die Ausgabe ueber das Master-Port nicht maskiert wird, da in einem solchen Fall wegen der ausbleibenden Interruptanmeldung das logische Geraet staendig besetzt bleibt und moeglicherweise die Task nicht weiterbearbeitet wird.
Eine Ausnahme bildet der Fall, wenn zwecks Uminitialisierung des logischen Geraetes alle Ports maskiert sind, sodass keine Ausgabe stattfindet.

* Variante DADB1

Die Ausgabemaske kann nur eine Laenge von 1 Byte besitzen. Werden Bloecke mit einer Laenge > 8 Byte maskiert ausgegeben, werden ab Port-Nr. 8 die Port unmaskiert bedient.

g) Standards:
Wird keine Adresse fuer den Ausgabepuffer im Ruf notiert, so gilt die in der Zuweisungs-Tabelle fest definierte Adresse.
Wird keine Maske oder Zeiger auf eine Maske im Ruf angegeben, so erfolgt die Ausgabe unmaskiert.
Erfolgt im Ruf keine Angabe eines Masters, so gilt die letzte Festlegung, wobei die erste Festlegung bei der Anfangsinitialisierung getroffen wird.
Wird im Ruf keine Satzanzahl definiert, so wird ein Satz ausgegeben.

Werden im Ruf keine Zeitkonstanten fuer die Zeitgeber definiert, so gilt die jeweils letzte Festlegung, wobei die erste Festlegung mit der Anlaufinitialisierung erfolgt.

h) Initialisierung:
Fuer die Treiber vom Typ DAD stehen die Standard-Initialisierungsroutinen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) zur Verfuegung. Der Anwender kann in der Initialisierungsroutine fuer das betreffende logische Geraet, deren Adresse in der Zuweisungs-Tabelle zu vermerken ist, die geeignete Standard-Routine als Unterprogramm aufrufen, nachdem die Parameterversorgung sichergestellt worden ist. Fuer den Typ DAD werden die PIO's (soweit vorhanden) auf "Byte-Ausgabe" initialisiert. Die Zeitgeber werden mit den in einer Tabelle "INIT-CTC" angegebenen Werten geladen, der Interruptvektor in die Zeitgeber aller Ports eingetragen sowie die festgelegte Adresse des Interruptsenders in die E/A-Tabelle des logischen Geraetes eingetragen.

Initialisierung DADE, DADD1, DADD2

Zur Initialisierung der Zeitgeber ist ueber Register HL eine Tabelle "INIT-CTC" folgender Form zu adressieren:

Byte-Nr.	Bedeutung
1	Interruptvektor
2	phys. Adresse Taktgeber (CTC-Kanal 0)
3	Zeitkonstante Taktgeber bzw. 0
4	Zeitkonstante Zeitgeber a bzw. 0
5	Zeitkonstante Zeitgeber b bzw. 0
6	phys. Adresse des Zeitgebers des Masters (phys. Adresse des Master-Ports + 5)

Es ist zu beachten, dass bei Angabe einer Zeitkonstante = 0 der betreffende CTC-Kanal nicht initialisiert wird.

Die initialisierte Impulsdauer auf dem Port, die durch die Laufzeit des zugeordneten Zeitgebers bestimmt ist, ergibt sich zu:

$$T_{imp} = t_s * 256 * T_t * T_z$$

Dabei bedeuten:

- t_s die Dauer des Systemtaktes
- T_t die Zeitkonstante des Taktgebers
- T_z die Zeitkonstante des Zeitgebers

Parameter:

< HL > : Adresse der Tabelle "INIT-CTC"

Initialisierung DADBI

Die Initialisierung erfolgt anhand des in der Zuweisungs-Tabelle definierten Portverzeichnisses. Dabei sind durch eine Maske "INIT-PIO" diejenigen Portadressen zu maskieren, die nicht auf einer PIO arbeiten. Dazu wird die entsprechende Bitposition mit "1" belegt, wobei Bit 0 des ersten Maskenbytes der ersten Portadresse fest zugeordnet ist.
Zur Initialisierung der Zeitgeber ist mit dem Register HL eine Tabelle "INIT-CTC" folgender Form zu adressieren:

Byte-Nr.	Bedeutung
1	Anzahl n der zu initialisierenden CTC
2	Interruptvektor
3	phys. Adresse Taktgeber a (CTC-Kanal 0)
4	Zeitkonstante Taktgeber a bzw. 0
5	Zeitkonstante Zeitgeber a. 1 bzw. 0
6	Zeitkonstante Zeitgeber a. 2 bzw. 0

7	phys. Adresse Taktgeber b
. . .	
m	phys. Adresse Taktgeber n
m+1	Zeitkonstante Taktgeber n bzw. 0
m+2	Zeitkonstante Zeitgeber n. 1 bzw. 0
m+3	Zeitkonstante Zeitgeber n. 2 bzw. 0
m+4	phys. Adresse des Zeitgebers des Masters (phys. Portadr. des Master-Ports + 5)

Es ist zu beachten, dass bei der Angabe einer Zeitkonstante=0 der betreffende CTC-Kanal nicht initialisiert wird.

Parameter:

< DE > = Adresse der Maske "INIT-PIO"
< HL > = Adresse der Tabelle "INIT-CTC"

3.8.4.3. Digitale Eingabe Statisch DES

a) Aufgaben:

Eingabe von digitalen Signalen im ungeteilten Verkehr in einen Eingabepuffer in folgenden, fuer das angesprochene logische Geraet bei der Systemgenerierung festgelegten Arten:

- Eingabe eines Bytes ueber ein in der Zuweisungs-Tabelle definiertes 8-Bit-Port (DESE).
- Eingabe eines Doppelbytes ueber ein in der Zuweisungs-Tabelle definiertes 16-Bit-Port (DESD).
- Eingabe eines Blockes von max. 255 Bytes ueber die Ports, deren physische Adressen in einem Port-Verzeichnis angegeben sind, dessen Adresse in der Zuweisungs-Tabelle angegeben ist (DESB).

b) Status: Ruf

c) Schreibweise:

```
[name] PIN  R=n,(DES=i[,ECA=(adr!symb)]
                     [,BOB=(adr!symb)]
```

[,(MASK=m!POM=(adr!symb))]
[,(TAKT=t!POT=(adr!symb))]
! PARA=(adr!symb))[;Kommentar]

Dabei bedeuten die Schluesselworte und Parameter:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \le n \le 2$
DES	–	digitale Eingabe statisch	
i	–	logische Geraetenummer	0 < i < 256
BOB	–	Adresse des Eingabepuffers	
ECA	–	Adresse Fehlerbyte	
MASK	–	Eingabe-Maske	
POM	–	Zeiger auf Eingabe-Maske	
TAKT	–	Zeitkonstante Taktgeber (nur fuer DESE, DESD)	
POT	–	Adresse der Liste "Taktzeiten" (nur fuer DESB)	
t	–	Wert der Zeitkonstanten	
PARA	–	Adresse des Parameterblockes	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	

d) Ruf-Nr: 7

e) Ruflaenge: 4-11 Byte

f) Wirkung:

Die Treiber vom Typ DES haben die Aufgabe, im Polling-Betrieb die in der Zuweisungs-Tabelle definierten Ports auszulesen, und die Information auf einen in der Zuweisungs-Tabelle oder im Ruf definierten Puffer auszugeben.

Fuer alle Systemkomponenten des logischen Geraetetypes DES werden bei Angabe einer Fehlerbyte-Adresse Fehlercodes in den adressierten Speicher uebertragen. Die moeglichen Fehlercodes sind:

0, 42H, 43H, 45H, 46H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

Fuer die Prozess-Steckkarten, die die Informationen getaktet

uebernehmen, koennen im Ruf die initialisierten Taktzeiten im Bereich von 0.1ms bis 25ms veraendert werden, indem eine neue Taktzeitkonstante bzw. die Adresse einer Liste von Taktzeiten folgenden Aufbaus im Ruf angegeben wird:

Byte-Nr.	Bedeutung
1	Anzahl n der Aenderungen
2	Port-Nr. 1
3	Zeitkonstante 1
. . .	
m-1	Port-Nr. n
m	Zeitkonstante n

Die eingestellte Taktzeit betraegt:

$$T_{Takt} = t_s * 256 * T_t$$

Dabei bedeuten:

T_{Takt} die Taktzeit

t_s die Dauer eines Systemtaktes

T_t die Taktzeitkonstante

Es ist zu beachten, dass auf den Prozess-Karten mit Takterzeugung jeweils nur ein Taktgeber vorhanden ist und sich deshalb Taktangaben fuer die verschiedenen Ports einer Karte auf den gleichen physischen Taktgeber beziehen koennen!
Ferner ist darauf zu achten, dass die Taktzeitkonstante bei Anwendung der Treiber DESE bzw. DESD im Ruf direkt, bei Anwendung der Treiber DESBI im Ruf durch Angabe einer Adresse der Liste von Taktzeiten angegeben werden muessen.

DESE Eingabe eines Bytes

Die Uebertragungsbreite betraegt 1 Byte, eine Maskendefinition wird ignoriert.

DESD Eingabe eines Doppelbytes

Die Uebertragung erfolgt aus zwei Ports mit unmittelbar aufeinanderfolgenden Adressen, eine Maskendefinition wird ignoriert.

DESBI Eingabe eines Datenblockes

Die Uebertragung erfolgt aus max. 255 Ports, deren Adressen in einem in der Zuweisungs-Tabelle adressierten Portverzeichnis enthalten sind. Ist eine Eingabemaske definiert, so werden die Ports nicht gelesen, deren Adressen in der entsprechenden Bitposition der Maske mit "1" maskiert sind. Dabei ist Bit 0 des ersten Maskenbytes der ersten Portadresse fest zugeordnet. Die Eingabe erfolgt in jedem Falle in einen lueckenlosen Pufferbereich entsprechend der Reihenfolge der im Portverzeichnis angegebenen physischen Port-Adressen.

* Variante DESB1

Die Eingabemaske kann nur die Laenge von 1 Byte besitzen. Werden Bloecke mit einer Laenge > 8 Byte eingegeben, werden die restlichen Ports unmaskiert gelesen.

g) Standards:
Wird keine Adresse fuer den Eingabepuffer im Ruf notiert, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse. Wird keine Maske oder Zeiger auf eine Maske im Ruf notiert, so erfolgt die Eingabe unmaskiert.
Werden im Ruf keine Zeitkonstanten fuer Taktgeber notiert, so gilt die jeweils letzte Einstellung, wobei die erste Einstellung mit der Anlauf-Initialisierung erfolgt.

h) Initialisierung:

Fuer die Treiber vom Typ DES stehen die Standard-Initialisierungsroutinen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Gerartetreiber) zur Verfuegung. Diese Routinen koennen als Unterprogramme in den Initialisierungsroutinen fuer die einzelnen logischen Geraete verwendet werden, nachdem ihre Parameterversorgung sichergestellt wurde.
Fuer den Typ DES werden die PIO's des logischen Geraetes auf "Byte-Eingabe" gesetzt und die Taktgeber (CTC-Kanal 0) mit einer wahlweisen Frequenz in Betrieb gesetzt.

Initialisierung DESE, DESD

Als Parameter ist die Adresse der Taktzeitkonstante anzugeben:

Parameter:

< HL > = Adresse der Taktzeitkonstante

Initialisierung DESBI

Die Initialisierung erfolgt anhand des in der Zuweisungs-Tabelle adressierten Portverzeichnisses. Dabei sind durch eine Maske "INIT-PIO" diejenigen Ports zu maskieren, die nicht auf einen PIO arbeiten (s.a. Initialisierung DAD). Weiterhin ist die Adresse einer Liste von Taktzeiten anzugeben; diese Tabelle besitzt folgenden Aufbau:

Byte-Nr:	Bedeutung
1	n = Anzahl der zu initialisierenden CTC
2	phys. Adresse Taktgeber a (CTC-Kanal 0)
3	Zeitkonstante Taktgeber a
. . .	
m	phys. Adresse Taktgeber n
m+1	Zeitkonstante Taktgeber n

Parameter:

< DE > = Adresse der Maske "INIT-PIO"

< HL > = Adresse der Liste "Taktzeiten"

3.8.4.4. Digitale Eingabe Statisch ueber Multiplexer DESM

a) Aufgaben:

Eingabe von digitalen Signalen im ungeteilten Verkehr ueber die Prozess-Steckkarten DEM in einen Eingabepuffer in folgenden, fuer das angesprochene logische Geraet bei der Generierung festgelegten Arten:

- Maskierbare Eingabe von 16 Byte ueber die Prozess-Steckkarte DEM, deren physische KES-Adresse in der Zuweisungs-Tabelle angegeben ist (DESME).
- Geblockte, unmaskierbare Eingabe ueber max. 255 Prozess-Steckkarten DEM (DESMB). Die physischen Adressen sind in einem KES-Verzeichnis enthalten, deren Adresse in der Zuweisungs-Tabelle angegeben ist.

b) Status: Ruf

c) Schreibweise:

```
[name] PIN  R=n,(DESM=i[,ECA=(adr!symb)]
                       [,BOB=(adr!symb)]
                       [,(MASK=m!POM=(adr!symb))]
                       [, WAIT]
                     ! PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
DESM	–	Digitale Eingabe statisch ueber Multiplexer	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adresse Fehlerbyte	
BOB	–	Adresse Eingabepufferbereich	
MASK	–	direktes Eingabemasken-Wort	
POM	–	Zeiger auf Eingabemaske	

WAIT   Aktivierung der WAIT-Funktion

PARA - Adresse des indirekten Parameterblocks

adr - absolute Adresse   0 < adr < FFFFH

symb - symbolische Adresse

d) Ruf-Nr.: 7

e) Ruflaenge: 4-11 Byte

f) Wirkung:

Die Treiber DESM uebertragen byteweise den aktuellen Speicher des DEM in den in der Zuweisungs-Tabelle oder im Ruf definierten Eingabepuffer.

Der Inhalt des DEM wird unabhaengig vom Systemtakt ca. alle 160ms vollstaendig aktualisiert.

Eine in dem Treiber DESM ausgeloeste Zeitsperre verhindert den Zugriff auf das logische Geraet vor Ablauf des Auffrischungszyklusses auf der (den) Prozess-Steckkarte(n). Dazu muss jedem logischen Geraet DESM bei der Anlaufinitialisierung eine Kontrollroutine zugeordnet werden. In dieser Kontrollroutine ist in HL die um 1 erhoehte Adresse der zum logischen Geraet zugehoerigen E/A-Tabelle (Adresse des Statusbytes) zu laden und zu der entsprechenden Routine zu verzweigen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber).

Die moeglichen Fehlercodes sind :

0, 42H, 43H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

DESME Eingabe von einer Prozess-Steckkarte DEM

In einen in der Zuweisungs-Tabelle oder im Ruf definierten Puffer wird der RAM-Inhalt der Prozess-Steckkarte, deren physische Adresse in der Zuweisungs-Tabelle angegeben ist, uebertragen.

Ist im Ruf eine Eingabemaske definiert, so werden die Informationen nicht uebertragen, deren zugeordneten Multiplexadressen (0-15) in der Eingabemaske mit "1" maskiert sind.

Dabei ist dem Bit 0 des ersten Maskenbytes die Multiplexadresse 0 fest zugeordnet. Die Eingabe erfolgt dabei in jedem Fall in einen lueckenlosen Pufferbereich.

DESMB Geblockte Eingabe ueber Prozess-Steckkarten DEM

In einen in der Zuweisungs-Tabelle oder im Ruf definierten Puffer werden die RAM-Inhalte der Prozess-Steckkarten DEM, deren physische Adressen in einem KES-Adressenverzeichnis enthalten sind, uebertragen (KES = Karteneinschub).
Dieses Verzeichnis besitzt formal den gleichen Aufbau wie ein Port-Verzeichnis. Die Reihenfolge der Eingabe erfolgt in aufsteigender Reihenfolge der Multiplexer-Adressen 0-15 und in der Reihenfolge der Notation der physischen Adressen im KES-Adressenverzeichnis.
Eine Eingabe-Maske wird ignoriert.

g) Standards:
Wird keine Adresse fuer den Eingabepuffer im Ruf notiert, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse.
Wird keine Maske oder Zeiger auf eine Maske im Ruf notiert, so erfolgt die Eingabe unmaskiert.

h) Initialisierung:
In der zum logischen Geraet gehoerenden Initialisierungsroutine ist in BC die Adresse der Kontrollroutine zu laden und zur zugehoerigen Initialisierungs-Routine zu verzweigen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber).

Parameter:
< BC > = Adresse Kontrollroutine

3.8.4.5. Digitale Eingabe Statisch mit Unterbrechung DESU

a) Aufgaben:
Verzoegerte Erfassung von Aenderungen digitaler Signale im

Interrupt-Betrieb in einen Eingabepuffer in einer der folgenden, fuer das angesprochene logische Geraet bei der Generierung festgelegten Arten:

- Ueberwachung von 8 Kanaelen ueber ein 8-Bit-Port, dessen phys. Adresse in der Zuweisungs-Tabelle angegeben ist (DEUE).
- Ueberwachung von 16 Kanaelen ueber ein 2 x 8-Bit-Port, dessen physische Adresse in der Zuweisungs-Tabelle angegeben ist (DEUD).
- Ueberwachung eines Blockes von max. 255 8-Bit-Ports, deren physische Adressen in einem Portverzeichnis angegeben sind (DEUB). Die Adresse dieses Portverzeichnisses ist in der Zuweisungs-Tabelle anzugeben.

b) Status: Ruf

c) Schreibweise:

```
[name] PIN  R=n,(DESU=i[,ECA=(adr!symb)]
                       [,BOB=(adr!symb)]
                       [,(MASK=m!POM=(adr!symb))]
                       [,(TAKT=t!POT=(adr!symb))] 1)
                       [,USER=(adr!symb)]
                       [,BOT=(adr!symb)]
                  ! PARA=(adr!symb))[;Kommentar]
```

1) TAKT nur fuer DEUE, DEUD
   POT nur fuer DEUB

Dabei bedeuten die Schluesselworte und Parameter:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
DESU	–	Digitale Eingabe statisch mit Unterbrechung	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adr. Fehlerbyte	
BOB	–	Adr. Eingabepufferbereich	
MASK	–	direktes Eingabemasken-Byte	
POM	–	Zeiger auf Eingabemaske	
USER	–	Adresse einer anwendereigenen ISR	

TAKT	–	Angabe einer neuen Taktzeit (nur fuer DEUE, DEUD)
t	–	Wert der Taktzeit
POT	–	Zeiger auf Angabe neuer Taktzeiten (nur fuer DEUB)
BOT	–	Bereich neuer Verzoegerungszeiten
PARA	–	Adresse des indirekten Rufparameterblockes
adr	–	absolute Adresse $0 < adr < FFFFH$
symb	–	symbolische Adresse

d) Ruf-Nr. 7

e) Ruf-Laenge: 4-17

f) Wirkung:

Die Treiber vom Typ DESU haben die Aufgabe, die in der Zuweisungs-Tabelle definierten Ports zum selbsttaetigen Auslesen im Interruptbetrieb zu aktivieren. Ein Interrupt wird dabei entweder sofort oder nach dem Ablauf des zugeordneten Zeitgebers angemeldet,welcher mit dem Auftreten einer mittels Loetprogrammierung fuer eine Gruppe von 8 Kanaelen ausgewaehlten Impulsflanke gestartet wurde. Waehrend der Laufzeit des Zeitgebers, die durch die eingestellte Taktzeit und die Zeitkonstante der Verzoegerung bestimmt ist,werden alle nachfolgenden Signalaenderungen erfasst und mit der Anerkennung des angemeldeten Interrupts auf einen in der Zuweisungs- Tabelle oder im Ruf definierten Puffer ausgegeben.

Neben der standardmaessigen Uebernahme der aktuellen Werte mit jedem Interrupt koennen die Interrupt-Serviceroutinen durch einen anwenderspezifischen Teil durch die entsprechende Angabe im Ruf erweitert werden. Der Eintritt in diesen Teil erfolgt im DI, wobei der Anwenderteil unter Verzicht auf die Verwendung von RAM-Speicherplaetzen, die nicht konkret einem logischen Geraet zugeordnet sind, auch unterbrechbar sein kann. Der Anwenderteil der ISR ist als normale Subroutine zu programmieren.

Eine Veraenderung der Taktzeiten des logischen Geraetes wird durch Angabe einer Taktzeitkonstante (DEUE, DEUD) bzw. durch Angabe eines Adresszeigers auf einen Bereich von Zeitkonstanten (DEUB) bewirkt.

Die Taktdauer ermittelt sich zu:

$$T_t = t_s * 256 * K_t$$

Dabei bedeuten:

T_t die Taktdauer

t_s die Dauer des Systemtaktes

k_t die Zeitkonstante des Taktgebers

Eine Veraenderung der Verzoegerungszeiten des logischen Geraetes erfolgt durch die Angabe der Adresse eines Bereiches von Verzoegerungszeitkonstanten im Ruf.
Die Verzoegerungszeit betraegt dabei:

$$T_i = T_t * K_i$$

Dabei bedeuten:

T_i die Verzoegerungszeit

T_t die Taktdauer

K_i die Verzoegerungszeitkonstante

Die programmierbaren Verzoegerungszeiten bewegen sich damit im Bereich von 0.1ms bis 6.6s.
Die erforderlichen Listen haben dabei folgenden Aufbau:

Byte-Nr:	Bedeutung
1	Anzahl n der Aenderungen
2	Port-Nr. 1
3	Zeitkonstante 1
...	

m-1 Port-Nr. n
m Zeitkonstante n

Fuer alle Systemkomponenten des logischen Geraetetypes DESU werden bei Angabe einer Fehlerbyte-Adresse im Ruf Fehlercodes in den adressierten Speicher uebertragen. Die moeglichen Fehlercodes sind:
0, 42H, 43H, 45H, 46H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

DEUE Eingabe eines Bytes

Die Uebertragungsbreite betraegt 1 Byte; eine Maskendefinition wird ignoriert. Die Taktzeit kann durch die direkte Angabe einer neuen Taktzeitkonstante veraendert werden.

DEUD Eingabe eines Doppelbytes

Die Uebertragung erfolgt aus 2 Ports mit unmittelbar aufeinanderfolgenden Adressen; eine Maskendefinition wird ignoriert. Die Taktzeit beider Ports kann durch die direkte Angabe einer neuen Taktzeitkonstante veraendert werden.
Die Verzoegerungszeiten fuer beide Ports koennen durch die Angabe der Adresse einer entsprechenden Liste von Verzoegerungszeiten veraendert werden.

DEUBI Eingabe eines Datenblockes

Die Uebertragung erfolgt aus max. 255 Ports, deren Adressen in einem in der Zuweisungs-Tabelle adressierten Portverzeichnis enthalten sind. Ist eine Eingabemaske definiert, so werden diejenigen Ports nicht gelesen, deren Adressen in der entsprechenden Bitposition der Maske mit "1" maskiert sind. Dabei ist Bit 0 des ersten Maskenbytes der ersten Adresse im Portverzeichnis fest zugeordnet.
Fuer jeden Port koennen die Taktzeit und die Verzoegerungszeit ueber Ruf veraendert werden, indem die Adressen der ent-

sprechenden Tabellen im Ruf angegeben werden. Beachte, dass der Taktgeber beide Ports einer Prozess-Karte bedient, sodass eine ungewollte Aenderung der Verzoegerungszeit des anderen Kanales moeglich ist, da diese von der Taktzeit abhaengig ist.

* Variante DEUB1

Eine Eingabemaske kann nur die Laenge von 1 Byte besitzen. Werden Bloecke mit einer Laenge > 8 Byte maskiert eingegeben, werden die restlichen Ports unmaskiert gelesen.

g) Standards:
Wird keine Adresse fuer den Eingabepuffer im Ruf notiert, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse. Wird keine Maske oder Zeiger auf eine Maske im Ruf notiert, erfolgt die Eingabe unmaskiert.
Wird im Ruf keine Zeitkonstante fuer den Taktgeber bzw. keine Adresse einer Liste "Taktzeiten" angegeben, so gilt die jeweils letzte Einstellung des Taktgebers, wobei das erste Einstellen mit der Anlaufinitialisierung erfolgt.
Wird im Ruf keine Adresse einer Liste "Verzoegerungen" fuer die Zeitgeber der Verzoegerungszeit angegeben, so gilt die jeweils letzte Einstellung der Zeitgeber, wobei das erste Einstellen mit der Anlaufinitialisierung erfolgt.

h) Initialisierung:
Fuer die Treiber vom Typ DESU stehen die Standard-Initialisierungsroutinen (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) zur Verfuegung.
Diese Routinen koennen als Unterprogramme in den Initialisierungsroutinen fuer die einzelnen logischen Geraete verwendet werden, nachdem deren Parameterversorgung sichergestellt wurde.
Fuer den Typ DEU werden die PIO's der Geraete auf "Byte-Eingabe" gestellt, die Taktgeber (CTC-Kanal 0) und die Zeitgeber fuer die Verzoegerungszeiten werden entsprechend einer Ta-

belle "INIT-CTC" gesetzt, wenn eine Verzoegerung der Interruptanmeldung erfolgen soll. Die Versorgungszeit dient zur Verzoegerung der Interruptanmeldung nach Auftreten der ersten Signalaenderung, deren Richtung fuer jedes Port durch Loetprogrammierung festgelegt ist. Diese Zeit ist so zu waehlen, dass alle moeglichen Folgeaenderungen der anderen Kanaele des Port bis zur Interruptanmeldung, die mit Ablauf des Zeitgebers erfolgt, erfasst werden koennen.
Die Laufzeit des Zeitgebers ermittelt sich zu:

$$T_{imp} = t_s * 256 * T_t * T_z$$

Dabei bedeuten:

t_s die Dauer des Systemtaktes

T_t die Zeitkonstante des Taktgebers

T_z die Zaehlkonstante des Zeitgebers

Initialisierung IDEUS

Fuer die sofortige Ausloesung eines Interrupts bei einer durch Loetprogrammierung festgelegten Signalaenderung an einem oder mehreren Eingaengen eines 8-Bit-Ports ist die Standard-Initialisierungsroutine IDEUS zu verwenden. Als Parameter ist der Interrupt- Vektor zu uebergeben.

Parameter:

< A > = Interrupt- Vektor

Initialisierung DEUE, DEUD

Zur Initialisierung der Zeitgeber ist eine Tabelle "INIT-CTC" folgender Form anzugeben.

Byte-Nr.	Bedeutung	
1	Interrupt-Vektor	
2	physische Adresse Taktgeber (CTC-Kanal 0)	
3	Zeitkonstante Taktgeber	bzw. 0
4	Zeitkonstante Zeitgeber 1	bzw. 0
5	Zeitkonstante Zeitgeber 2	bzw. 0

Die Angabe einer Zeitkonstante=0 verhindert die Initialisierung des entsprechenden Kanals.

Parameter:

< HL > = Adresse Tabelle "INIT-CTC"

Initialisierung DEUB

Die Initialisierung erfolgt anhand des Portverzeichnisses des logischen Geraetes. Dabei werden durch eine Maske "INIT-PIO" diejenigen Portadressen maskiert, die nicht auf einer PIO arbeiten.
Zur Initialisierung der Zeitgeber ist eine Tabelle "INIT-CTC" folgender Form anzugeben:

Byte-Nr.	Bedeutung	
1	n = Anzahl der zu initialisierenden CTC's	
2	Interruptvektor	
3	physische Adresse Taktgeber 1 (CTC-Kanal 0)	
4	Zeitkonstante Taktgeber 1	bzw. 0
5	Zeitkonstante Zeitgeber 1.1	bzw. 0
6	Zeitkonstante Zeitgeber 1.2	bzw. 0
7	physische Adresse Taktgeber 2	
. . .		
m	physische Adresse Taktgeber n	
m+1	Zeitkonstante Taktgeber n	bzw. 0
m+2	Zeitkonstante Zeitgeber n.1	bzw. 0

m+3 Zeitkonstante Zeitgeber n.2 bzw. 0

Die Angabe einer Zeitkonstanten=0 verhindert die Initialisierung des entsprechenden Kanals.

Parameter:

< DE > = Adresse der Maske "INIT-PIO"

< HL > = Adresse der Liste "INIT-CTC"

3.8.4.6. Digitale Eingabe Dynamisch DED

a) Aufgabe:

Erfassung von Impulsen ueber ein 8-Bit-Port, dessen phys. Adresse in der Zuweisungs-Tabelle angegeben ist (DEDE).

b) Status: Ruf

c) Schreibweise:

```
[name] PIN  R=n,(DED=i[,ECA=(adr!symb)]
                     [,BOB=(adr!symb)]
                     [,USER=(adr!symb)]
                     [,MEMORY]
                   ! PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
DED	–	Digitale Eingabe dynamisch	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adresse Fehlerbyte	
BOB	–	Adresse Eingabepufferbereich	
USER	–	Adresse einer anwendereigenen ISR	
MEMORY		Speicheranweisung fuer alle einlaufenden Impulse	
PARA	–	Adresse des indirekten Rufparameterblocks	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	

d) Ruf-Nr. 7

e) Ruflaenge: 4-11 Byte

f) Wirkung:

Der Treiber vom Typ DED hat die Aufgabe, den in der Zuweisungs-Tabelle definierten Port zum selbsttaetigen Ausloesen im Interrupt-Betrieb zu aktivieren. Ein Interrupt wird dabei mit dem Auftreten einer LOW/HIGH-Flanke fuer die Prozesskarten mit TTL-Eingaengen, sonst mit einer HIGH/LOW-Flanke an einem der Kanaele angemeldet. In der Interrupt-Service-Routine (ISR) wird standardmaessig die zum Zeitpunkt der Anerkennung der Interrupts bestehende Portbelegung in den in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer uebertragen, wobei die interruptausloesende Signalflanke durch ein LOW-Signal abgebildet wird.

Durch die optionale Angabe der MEMORY-Funktion wird bewirkt, dass die Portbelegung disjunktiv mit dem aktuellen Pufferinhalt verknuepft wird. Es ist darauf zu achten, dass bei eingeschalteter MEMORY-Funktion der Puffer zum geeigneten Zeitpunkt vom Anwender auf FFH, geloescht werden muss!

Die ISR kann wahlweise durch einen anwendereigenen Teil erweitert werden, indem die entsprechende Adressangabe im Ruf vorgenommen wird. Der Eintritt in diesen Anwenderteil erfolgt im DI, wobei bei entsprechender Gestaltung dieses Unterprogramm unterbrechbar sein kann. Der anwendereigene Teil der ISR darf die in EIEX fuer ISR zugelassenen Rufe enthalten. Der anwendereigene Teil ist als Subroutine zu programmieren.

Bei Angabe einer Fehlerbyte- Adresse im Ruf koennen folgende Fehlerbytes in den adressierten Speicher uebertragen werden 0, 42H, 43H (siehe Anlage 3, Felerausschriften und Systemnachrichten EIEX 1521).

DEDE Impulseingabe ueber 1 Byte

Die Ueberwachung erfolgt ueber eine Breite von 8 Kanaelen (1 Byte).

g) Stanaards:
Wird keine Adresse des Eingabe-Puffers angegeben, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse.

h) Initialisierung:
Die entsprechende Standard-Initialisierungs-Routine (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) kann als Unterprogramm in der Initialisierungsroutine fuer das logische Geraet verwendet werden. Die Standardroutine setzt den in der Zuweisungs-Tabelle definierten Port auf "Bit-Eingabe". Gleichzeitig werden alle Kanaele zur Interruptbildung zugelassen, wobei diese bei HIGH/LOW-Flanke erfolgt. Der angegebene Vektor wird in den zum logischen Geraet zugehoerigen Port eingetragen.

Parameter:

< A > = Interrupt-Vektor

3.8.4.7. Digitale Ein/Ausgabe ueber die Prozess-Steckkarte DEAS DAC / DEC

a) Aufgaben:
Ein-und Ausgabe von statischen digitalen Signalen im Interruptbetrieb in folgenden, fuer das angesprochene logische Geraet bei der Generierung festgelegten Arten:
- Ausgabe von einem Byte ueber ein 8-Bit-Port, dessen phys. Adresse in der Zuweisungs-Tabelle angegeben ist (DACE).
- Eingabe von einem Byte ueber ein 8-Bit-Port, dessen phys. Adresse in der Zuweisungs-Tabelle angegeben ist (DECE).
- Ausgabe von max. 255 Byte ueber ein 8-Bit-Port, dessen physische Adresse in der Zuweisungs-Tabelle angegeben ist (DACB).
- Eingabe von max. 255 Byte ueber ein 8-Bit-Port, dessen phys. Adresse in der Zuweisungs-Tabelle angegeben ist (DECB).
- Bidirektionaler Verkehr (DACB, DECB)

b) Status: Ruf

c) Schreibweise:

```
([name] POUT  R=n,(DAC=i[,ECA=(adr!symb)]
                        [,BOB=(adr!symb)]
                        [,SIZE=m]
                        [,WAIT]
                   ! PARA=(adr!symb))[;Kommentar]
![name]  PIN  R=n,(DEC=i[,ECA=(adr!symb)]
                        [,BOB=(adr!symb)]
                        [,USER=(adr!symb)]
                   ! PARA=(adr!symb)))[;Kommentar]
```

Dabei bedeuten:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
DAC	–	Digitale Ausgabe ueber die Prozess-Steckkarte DEAS	
DEC	–	Digitale Eingabe ueber die Prozess-Steckkarte DEAS	
i	–	logische Geraete-Nr.	$0 < n < 256$
BOB	–	Adresse des Pufferbereiches	
SIZE	–	Blocklaenge	
m	–	Wert der Blocklaenge	$0 < m < 256$
USER	–	Adresse eines anwendereigenen Teils der ISR	
WAIT		Aktivierung der WAIT-Funktion	
PARA	–	Adresse des indirekten Rufparameterblockes	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	

d) Ruf-Nr: 7

e) Ruf-Laenge: 4-11 Byte

f) Wirkung:

Fuer die Durchfuehrung eines Interrupt-Betriebs sind fuer die Prozess-Steckkarte DEAS die Treiber vom Typ DAC bzw. DEC vorgesehen.

Der Treiber DACE dient der Ausgabe eines Bytes; ueber die Handshake-Leitung wird die Abnahme dieses Bytes signalisiert. Ist die WAIT-Funktion aktiviert, wird der Ruf DACE bis zur Abnahme des gesendeten Bytes nicht verlassen.
Der Treiber DECE dient der Parameterversorgung der zugehoerigen Interrupt-Service-Routine, die mit einer Eingabe-Anforderung ueber die Handshake-Leitung im Byte-Betrieb oder mit einer Interruptausloesung im Bit-Betrieb gestartet wird. Wurde im Ruf die Adresse einer Anwender-ISR angegeben, wird nach der Eingabe eines Bytes in diese Routine verzweigt. Der Eintritt in die Anwender-ISR erfolgt im DI; der Austritt aus dieser Routine erfolgt ueber RET.

Der Treiber DACB hat die Aufgabe, die Uebertragung eines Blockes einzuleiten. Zunaechst wird die Blocklaenge ausgegeben, wobei Blocklaengen von 0-255 Byte moeglich sind. Die Ausgabe des Datenblockes uebernimmt die zugeordnete System-ISR.

Der Treiber DECB hat die Aufgabe, die zugeordnete ISR mit den notwendigen Informationen zu versorgen, die zur Uebernahme eines Datenblockes notwendig sind. Die ISR liest auf Aufforderung von der Handshake-Leitung hin zunaechst die gesendete Blocklaenge, um anschliessend interruptgesteuert den gesamten Datenblock auf den vereinbarten Eingabepuffer zu schreiben. Wurde im Ruf die Adresse eines anwenderspezifischen Teils der ISR angegeben, so wird nach Abschluss der Eingabe dorthin verzweigt; die Rueckkehr aus dieser Routine erfolgt ueber RET.
Beim bidirektionalen Betrieb erfolgt aufgrund der konstruktiven Gestaltung der Prozesskarte DEAS die Steuerung des Verkehrs nur unter Benutzung der Handshakesignale des Port A, (Voraussetzung dafuer ist eine direkte Verbindung der Signale ARDY und BSTB am Prozess-Stecker der Karte!).

Der Ein- und Ausgabe-Kanal muss zwei unterschiedlichen logischen Geraeten zugeordnet sein, die jedoch mit einer gemeinsamen E/A-Tabelle arbeiten. Dabei richtet sich die Platzreservierung fuer diese Tabelle nach dem groesseren Bedarf des Eingabetreibers.
Alle Geraetetreiber fuer die Prozess-Steckkarte DEAS koennen bei Angabe einer Fehlerbyte- Adresse im Ruf folgende Fehlercodes in den adressierten Speicher uebertragen:
0, 42H, 43H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

g) Standards:
Wird im Ruf keine Adresse eines Puffers angegeben, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse.

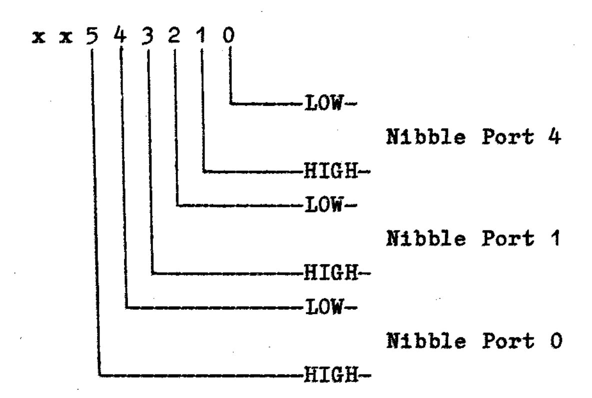
h) Initialisierung:
Fuer die Prozesskarte DEAS stehen die Standard-Initialisierungs-Routinen fuer den wahlfreien Verkehr und fuer den bidirektionalen Verkehr zur Verfuegung. Bei der wahlfreien Initialisierung koennen die drei E/A-Port im Byte-Ein/Ausgabe-Mode (Port 0 bis 1 wahlweise im Interrupt-Betrieb) oder im Bit-Mode betrieben werden, wobei entsprechend der konstruktiven Loesung die I/O-Zuordnung nur halbbyteweise erfolgen kann.
Bei der Initialisierung fuer den bidirektionalen Betrieb sollte Port 4 im Bit-Mode betrieben werden; Port 1 wird im Bit-Mode mit gesperrtem Interrupt, Port 0 im bidirektionalen Mode betrieben.
Die Festlegung der Richtung des Datentransfers wird im Statusbyte festgelegt. Eine Ausnahme bildet der Verkehr mit Port 0 im bidirektionalen Betrieb; hierbei werden die auf der Prozesskarte eingesetzten Treiberstufen automatisch durch ARDY umgesteuert.
Soll ein Port im Byte-Betrieb arbeiten, so muessen die Angaben fuer die Transferrichtungen fuer das Port im Statusbyte einheitlich sein; die Betriebsart wird aus der Richtungsangabe des Statusbyte abgeleitet.

Um einen definierten Betrieb der Prozesskarte unter EIEX 1521 zu gewaehrleisten, wird die Karte beim ersten Ansprechen in ihrer Gesamtheit initialisiert, wobei die Ports verschiedenen logischen Geraeten zugeordnet sein koennen. Alle mit dieser Prozesskarte arbeitenden logischen Geraete werden gleichzeitig als anlaufinitialisiert gekennzeichnet. Die Reihenfolge der Angabe dieser logischen Geraete ist beliebig.
Zur Initialisierung fuer wahlfreien Verkehr muss eine Initialisierungstabelle folgender Form angegeben werden:

Status-Byte
Initialisierungsparameter Port 4
Initialisierungsparameter Port 1
Initialisierungsparameter Port 0
Anzahl n der logischen Geraete
logisches Geraet Nr. a
. . .
logisches Geraet Nr. n

Soll ein logisches Geraet im Wartebetrieb arbeiten, so ist das Bit 7 der zugeordneten logischen Geraetenummer auf Eins zu setzen.

Die Zuordnung der Richtungsfestlegung im Statusbyte ist dabei folgende:

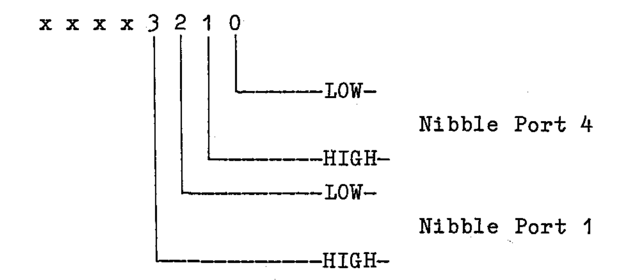


wobei die Belegung mit "1" die Eingabe,
die Belegung mit "0" die Ausgabe initialisieren.

Zur Initialisierung fuer den bidirektionalen Betrieb ist eine Initialisierungs-Tabelle folgender Form anzugeben:

Status-Byte
Initialisierungsparameter Port 4
Interruptvektor
Anzahl n der logischen Geraete
logisches Geraet Nr. a
. . .
logisches Geraet Nr. n

Die Zuordnung der Richtungsfestlegung im Statusbyte ist dabei folgende:



wobei die gleichen Festlegungen wie oben gelten.
Fuer den Aufbau der Initialisierungsparameter eines Port gelten folgende Festlegungen:

1) Fuer die Initialisierung fuer Byte-Betrieb ist
   - im Interrupt-Betrieb die Angabe eines Interrupt-Vektors
   - sonst die Angabe des Steuerwortes 03H (Intrrt-e)

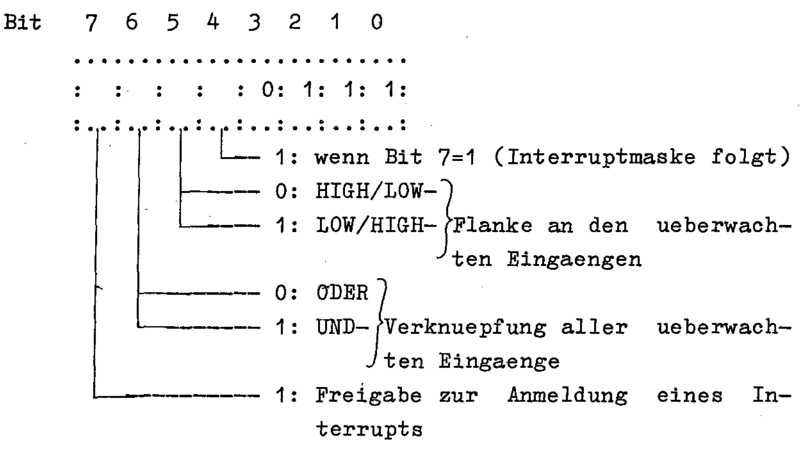
   erforderlich. Beachte, dass Port 4 nicht im Interrupt-Betrieb arbeiten kann!
2) Fuer die Initialisierung fuer Bit-Mode sind folgende Angaben zu notieren:

   Byte1: Interrupt-Control-Byte
   ist im Interrupt-Control-Byte Bit 7 = "1" (Interrupt-Freigabe), dann sind zusaetzlich anzugeben:

   Byte2: Interrupt-Maske (wobei im Interrupt-Control-Byte Bit 4 = "1" sein muss)

   Byte3: Interrupt-Vektor

Das Interruptkontrollbyte hat dabei folgenden Aufbau:



Die ueberwachten Eingaenge in der Interruptmaske werden mit "0" gekennzeichnet.

Jede andere denkbare Parameterkombination gilt als Fehler und fuehrt bei generierter Systemnachrichtenorganisation zur Ausgabe einer Fehlermeldung; darueber hinaus wird die Anzahl der Initialisierungsfehler in der Arbeitszelle ARBE aufsummiert.

Parameter:

< HL > = Adresse der Initialisierungs-Tabelle

3.8.4.8. Impulsausgabe ueber die Prozess- Steckkarte IA IA

a) Aufgabe:

Ausgabe von Digitalwerten aus einem in der Zuweisungs-Tabelle oder im Ruf definierten Ausgabepuffer auf die Prozess-Steckkarte IA zur Umsetzung in eines der folgenden, bei der Generierung festgelegten digitalen Signale.

- Impulszeitsignal entsprechend einer 1-Byte Zeitkonstante und einem Vorzeichen (IAZTE)
- Impulszeitsignal entsprechend eines 2-Byte langen Bitmusters und eines Vorzeichens wahlweise mit einer Anlaufimpulsfolge (IAZLE)

b) Status: Ruf

c) Schreibweise:

```
[name] POUT  R=n,(IA=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,USER=(adr!symb)]
                      [,SERO=(adr!symb)]
                      [,WAIT]
                      [,BREAK]
                    ! PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
IA	–	Prozess-Steckkarte	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adresse-Fehlerbyte	
BOB	–	Adresse des Ausgabepuffers	
USER	–	Adresse einer anwenderspezifischen ISR	
SERO	–	Adresse einer Statusfehlerroutine	
WAIT	–	Aktivierung der WAIT-Funktion	
BREAK	–	Zugriff auf ein besetztes logisches Geraet	
PARA	–	Adresse eines indirekten Rufparameterblockes	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	

d) Ruf-Nummer: 7

e) Ruf-Laenge: 4–13 Byte

f) Wirkung:

Bei der Eroeffnung des E/A-Verkehrs wird das logische Geraet als besetzt gekennzeichnet und der Geraetestatus der angesprochenen Prozess-Steckkarte untersucht. Liegt der Geraetestatus "Ausser Betrieb" vor, so werden folgende Aktivitaeten unternommen.

- Ausgabe einer Systemmeldung, bei generierter Systemnachrichtenorganisation,

- Ansprung einer anwendereigenen Statusfehlerroutine. Dabei wird in A die physische Kanaladresse des angesprochenen Kanals der Prozess-Steckkarte uebergeben.
  Der Aussprung aus dieser Statusfehlerroutine erfolgt ueber RET, der Inhalt von A bestimmt bei Ruecksprung, ob der E/A-Verkehr abgebrochen (A=0) oder fortgefuehrt (A=0) werden soll.

Die Beendigung des E/A-Verkehrs wird von der Prozess-Steckkarte mit der Anmeldung eines Interruptes angezeigt.

In dem vom Anwender zu programmierenden Eintritt in die System-ISR ist die Registerrettung aufzurufen, in HL die Adresse der zugehoerigen E/A-Tabelle zu laden und zur ISR (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess - Geraetetreiber) zu verzweigen.

Wurde im Ruf die Adresse einer anwendereigenen ISR angegeben, wird in der System-ISR der Geraete-Status noch einmal abgefragt. Liegt jetzt der Status "Ausser Betrieb" vor, verzweigt die ISR in diesem Fall zu der angegebenen Adresse. Der Ruecksprung aus der anwendereigenen ISR ins Betriebssystem erfolgt ueber RET. Hier wird das logische Geraet als wieder frei gekennzeichnet. Da sich die Impulsausgaben ueber einen laengeren Zeitraum erstrecken koennen, ist es bei Angabe der BREAK-Funktion moeglich, auf ein besetztes logisches Geraet zuzugreifen.

Das Zeitregime der Impulsausgabe wird auf der Prozess-Steckkarte durch zwei Taktgeber bestimmt, die bei der Anlaufinitialisierung wahlweise zu einem Taktgeber zur Erzielung grosser Taktzeiten gekoppelt werden koennen. Der Takt kann dabei wahlweise entsprechend der Anfangsinitialisierung vom internen Systemtakt oder von einem externen Taktsignal abgeleitet werden.

Die Geraetetreiber zur Bedienung der Prozess-Steckkarte IA koennen bei Angabe einer Fehlerbyteadresse im Ruf folgende Fehlercodes in den adressierten Speicher uebertragen:
0, 24H, 42H, 43H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

IAZTE Ausgabe eines Impulszeitsignales

Die Ausgabedaten fuer jedes logische Geraet bestehen aus 2 Byte mit folgender Struktur

Byte	Bit 7	6	5	4	3	2	1	0	
1	x	x	x	x	x	x	x	VZ	(Vorzeichen)
2	Zeitkonstante (zulaessiger Wertebereich der Zeitkonstante 1-0FFH)								

Durch den Treiber werden diese Informationen an der zum angesprochenen Kanal zugeordneten Stelle auf der Prozess-Steckkarte eingetragen.
Die Dauer des Impulszeitsignales betraegt dabei

$$T_Z = T_t * \text{Zeitkonstante}$$

wobei T_t die eingestellte Taktzeit bedeutet.

IAZLE Ausgabe eines Impulszaehlsignales

Die Ausgabedaten fuer jedes logische Geraet bestehen im Standardfall aus 3 Byte mit folgender Struktur:

Byte	Bit 7	6	5	4	3	2	1	0	
1	0	0	0	0	0	0	0	VZ	(Vorzeichen)
2	Impulsanzahl							LOW	
3								HIGH	

(zulaessige Impulszahl 0001...0FFFFH, wobei 0=65536)

Durch den Treiber werden diese Informationen an die zum angesprochenen Kanal zugeordnete Stelle eingetragen.
Durch die Prozess-Steckkarte wird die im Ausgabepuffer angegebene Anzahl von Impulsen mit der eingestellten Taktfrequenz ausgegeben.
Fuer die Realisierung von Anlaufprozessen koennen waehrend der Impulsanzahlausgabe maximal 127 sogenannte Aenderungspunkte vorgesehen werden, in denen die Ausgabefrequenzen nach je maximal 256 Impulsen geaendert werden koennen. Die Ausgabedaten besitzen dann folgende Struktur:

Byte	Bit 7 6 5 4 3 2 1 0		
1	n	VZ	(Vorzeichen und
2		LOW	Angabe der An-
	Impulsanzahl	----	zahl der Aende-
3		HIGH	rungspunkte n)
4	1. Takt-Steuerwort		
5	1. Taktzeitkonstante		
6	1. Anlaufimpulszahl		
	.		
	.		
	.		
m	n.tes Taktsteuerwort		
m+1	n.te Taktzeitkonstante		
m+2	n.te Anlaufimpulszahl		

Nach Ablauf der n-ten (letzten) Anlaufimpulszahl wird die Takterzeugung wieder mit den Initialisierungsparametern geladen. Dabei sind folgende Steuerworte fuer die Takterzeugung zulaessig:

- 55H externer Takt
- 05H interne Takterzeugung Vorteiler 16
- 25H interne Takterzeugung Vorteiler 256

(es ist nur eine getrennte Takterzeugung zulaessig (siehe Pkt. h, Initialisierung))

g) Standards:

Wird keine Adresse des Eingabepuffers im Ruf notiert, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse. Werden keine Adressen fuer eine Statusfehler-Routine bzw. anwendereigene ISR im Ruf notiert, werden keine anwenderspezifischen Reaktionen auf Statusfehler eingeleitet.

h) Initialisierung:

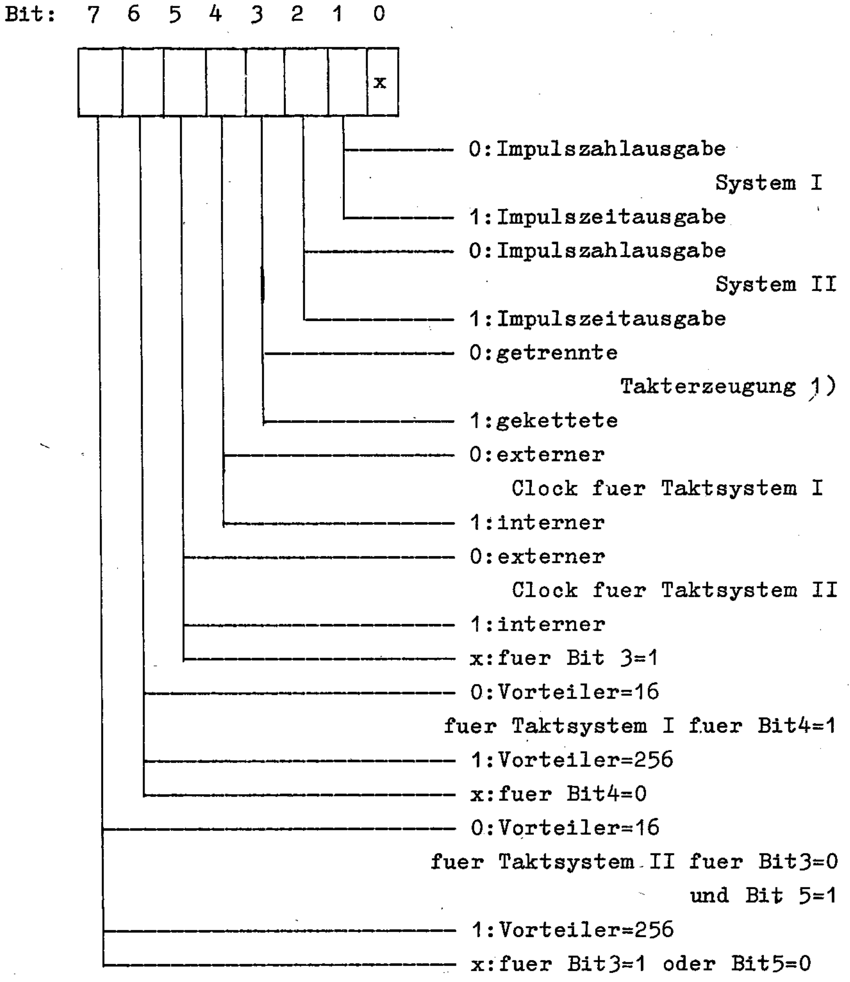
Fuer die Treiber vom Typ IA steht die Standard-Initialisierungsroutine (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) zur Verfuegung.

Durch die Initialisierungsroutinen erfolgt die Initialisierung der Schnittstellenschaltkreise PIO und 2 x CTC. Damit erfolgt die Strukturierung der Ausgabekanaele, die Strukturierung der Takterzeugung und Festlegung des Zaehltaktes. Da die Prozesskarte in ihrer Gesamtheit beim Systemanlauf initialisiert wird, werden die Initialisierungsparameter fuer alle (max. 4) logischen Geraete in einer Initialisierungstabelle zusammengefasst. Die behandelten logischen Geraete werden als anlaufinitialisiert gekennzeichnet.

Die ueber <HL> zu adressierende Initialisierungstabelle hat folgenden Aufbau:

Byte-Nr.	Bedeutung
1	Struktur-Wort
2	Takt-ZK0 (System I)
3	Takt-ZK1 (System II)
4	Vektor 0 (System I)
5	Vektor 1 (System II)
6	Anzahl der mit der Prozess-Steckkarte arbeitenden logischen Geraete
7	logische Geraete-Nr. 1
.	
.	
m	logische Geraete-Nr. n

Das Strukturwort besitzt dabei folgenden Aufbau:



wobei x eine gleichgueltige Belegung darstellt

1) Bei der Strukturierung ist zu beachten, dass Bit3=0 sein muss, wenn eine Impulszahlausgabe mit Aenderungspunkten moeglich sein soll.

3.8.4.9. Bedienung der Prozess-Steckkarte UIZ

a) Aufgabe:

Maskierbare Eingabe von Informationen von einer Prozesskarte UIZ in den Eingabepuffer, wobei die uebertragenen Informationen in einer der folgenden, beim Systemanlauf durch Initialisierung festgelegten Betriebsarten erzeugt werden.

- Impulszaehler
- Frequenzmesser
- Zeitimpulsgeber
- Zeitmesser

b) Status: Ruf

c) Schreibweise:

```
[name] PIN  R=n,(UIZ=i [,ECA=(adr!symb)]
                       [,BOB=(adr!symb)]
                       [,(MASK=m!POM=(adr!symb))]
                       [,POT=(adr!symb)]
                       [,USER=(adr!symb)]
                     ! PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten:

R	- Registerrettung	
n	- Registerrettungsart	$0 \le n \le 2$
UIZ	- Prozess-Steckkarte	
i	- logische Geraete-Nummer	$0 < i < 256$
ECA	- Adresse des Fehlerbytes	
BOB	- Adresse des Eingabepuffers	
MASK	- Eingabe-Maske	
POM	- Zeiger auf Eingabe-Maske	
POT	- Zeiger auf eine Liste "Zeitkonstanten"	
USER	- Adresse eines anwenderspezifischen Teils der ISR	
PARA	- Adresse des indirekten Rufparameterblockes	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

d) Ruf-Nr: 7

e) Ruf-Laenge: 4 -15 Byte

f) Wirkung:

Eingabe der Zaehlerstaende von maximal 4 CTC-Kanaelen mit einer Kapazitaet von jeweils 256 Bit.

Die Eingabe erfolgt von der Prozess-Steckkarte in einen lueckenlosen Pufferbereich. Wird im Ruf eine Eingabemaske oder ein Zeiger auf eine Eingabemaske angegeben, werden nur die Informationen der Kanaele gelesen, deren Maskenbit gleich Null ist. Den Kanaelen 0...3 sind die Maskenbits 0...3 fest zugeordnet.

Ist Bit 4 der Eingabemaske gleich 0, wird fuer jeden interruptsendenden Kanal der zur Kapazitaetserweiterung dienende Ueberlauf-Zaehler inkrementiert.

Wird im Ruf die Adresse einer Liste "Zeitkonstanten" angegeben, dann werden diese zu den entsprechenden Kanaelen ausgegeben. Die Liste der Zeitkonstanten muss folgenden Aufbau besitzen:

Byte-Nr.	Bedeutung
1	n = Anzahl der Aenderungen (1...4)
2	1.Kanal-Nummer (0...3)
3	Zeitkonstanten fuer 1.Kanal
.	
.	
2n	n-te Kanal-Nummer (0...3)
2n+1	Zeitkonstante fuer n-ten Kanal

Bei der Abfrage geketteter Kanaele muss eine Ergebnisverfaelschung als Folge eines zwischenzeitlichen Ueberlaufs ausgeschlossen werden. Deshalb wird bei diesen Varianten getestet, ob der Pufferinhalt des/der zuerst abgefragten niederwertigen Kanals/Kanaele (0 und/oder 2) groesser als 3 ist. Faellt der Test negativ aus, wird die Abfrage wiederholt und dieser Wert ebenfalls getestet. Bei erneutem negativen Testergebnis ist ein Ueberlauf ausgeschlossen, anderenfalls wird der Eingabe-

puffer mit den Daten der zweiten Abfrage ueberschrieben. Bei geketteten Kanaelen muss die Eingabemaske fuer alle geketteten Kanaele gleich sein, die Zeitkonstanten 1-3 sind fuer den niederwertigen Kanal nicht zugelassen. Zur Durchfuehrung des in den Betriebsarten "Impulszaehler" und "Zeitmesser" erforderlichen Ueberlauftests wird bei der Erstinitialisierung die Kettenkennzeichnung aus der Initialisierungstabelle in das Status-Byte kopiert.

Zur Bedienung der von den CTC-Kanaelen angemeldeten Interrupts sind entsprechende System-ISR's (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) vorhanden. Im projektabhaengigen Eintritt in die System-ISR, muss die generierte Registerrettungsroutine gerufen, die Adresse der E/A-Tabelle des interruptanfordernden logischen Geraetes nach HL sowie die Nummer des interruptanfordernden CTC-Kanals (0...3) in C eingezogen werden, um anschliessend zur System-ISR zu verzweigen.

Die System-ISR dient der maskierbaren Eingabe der Zaehlerstaende in die zugeordneten Eingabepuffer (speziell in Betriebsart "Frequenzmesser"). Eingabemaske und Pufferadresse werden der E/A- Tabelle entnommen. Ist Bit 4 der Eingabemaske gleich Null, wird der in der E/A- Tabelle enthaltene Ueberlaufzaehler des interruptsendenden Kanals inkrementiert (Zaehlkapazitaets-Erweiterung). Ausserdem erfolgt eine maskierte Neuinitialisierung der PIO-Ports und CTC-Kanaele (speziell fuer Betriebsarten "Frequenzmesser" und "Zeitimpulsgeber"). Die dazu benoetigten Initialisierungsmaske, Datenworte der PIO-Ports und Steuerworte sowie Zeitkonstanten der CTC-Kanaele werden der E/A-Tabelle entnommen. Ist die zugeordnete Bitposition der Initialisierungsmaske gleich "1" gesetzt, erfolgt keine Neuinitialisierung. Bit 7 der Initialisierungsmaske wird nicht ausgewertet.

Die System-ISR kann durch einen anwenderspezifischen Teil ergaenzt werden. Wurde in einem der vorangegangenen Rufe die Adresse einer Anwender-ISR angegeben, so wird zu dieser Routine verzweigt. Der Eintritt in die Anwender-ISR erfolgt im DI, der Ruecksprung in das System muss ueber RET erfolgen.

- Variante Impulszaehler

Der UIZ besitzt vier Kanaele mit einer Kapazitaet von jeweils 256 Bit, die als Rueckwaertszaehler arbeiten. Der verwendete CTC zaehlt von dem per Programm eingegebenen Wert bis Null und springt danach sofort auf den Voreinstellwert zurueck. Bei Nulldurchgang wird der Ausgang ZC/TO des betreffenden Kanals aktiv, ausserdem wird ein Interrupt generiert, falls dieser freigegeben war. Die vier Kanaele des CTC werden durch Variation der Adressbits AB 0 und AB 1 (00..11) angesprochen. Die Kettung der einzelnen Zaehler ist bis zu einer Laenge von 32 Bit moeglich, dabei wird der ZC/TO-Ausgang ueber die Programmierung an den CLK/TRG-Eingang des Zaehlers mit der naechsthoeheren Adresse angeschlossen. Der gekettete Zaehler wird ueber den Zaehlereingang mit der niedrigsten Adresse inkrementiert.
Bei der Kettung mehrerer Zaehler ergibt sich der Voreinstellwert aus dem Produkt der Voreinstellwerte der geketteten Kanaele.

- Variante Frequenzmesser

Die geketteten Kanaele 0 und 1 dienen der Erzeugung einer modulinternen Zeitbasis. Der Start des Zeitimpulsgebers erfolgt ueber Kanal 0. Der Zeitimpulsgeber kann vom Prozess, per Programm oder von einem Koppelbussignal (ZSK) durch eine H/L-Flanke an CLK/TRG 0 gestartet werden.
Waehrend der Laufzeit des Impulsgebers haelt ein bistabiler Multivibrator (BM) die Eingaenge der Kanaele 3 und 4 (2 x 8 Bit) bzw. von Kanal 3 (1 X 16 Bit) offen, an die eine bzw. 2 Impulsquellen angeschlossen sind. Am Ende des Zeitimpulses werden die Eingaenge gesperrt, und ueber Kanal 1 kann ein Interrupt ausgeloest werden, der der CPU das Ende der Messung anzeigt. Die CPU liest danach die Kanaele 3 und 4 und kann dann aus diesen Daten Frequenz oder Impulsrate der angeschlossenen Quellen ermitteln.

Die Messzeit ergibt sich aus dem Produkt der Voreinstellwerte der Kanaele 0 und 1 und kann zwischen 0,0065 ms (praktisch nicht verwendbarer Minimalwert) und 6,9 s variiert werden.

- Variante Zeitimpulsgeber

Das Zeitsignal wird mit Hilfe der geketteten Kanaele 0 und 1 erzeugt. Der Start erfolgt ueber Kanal 0. Der Zeitimpulsgeber kann vom Prozess, per Programm oder durch ein Koppelbussignal (ZSK) durch eine H/L-Flanke an CLK/TRG 0 gestartet werden. Waehrend der zwischen 0,0065 ms und 6,9 s programmierbaren Laufzeit wird ein potentialfreier elektronischer Kontakt geschlossen, der vom Prozess aus zugaenglich ist sowie ein Signal ueber ZAK zum Koppelbus ausgegeben.
(Am Ausgang ZAK liegt der offene Kollektor eines in Basisschaltung betriebenen Transistors).
Soll der Start eines Frequenzmessers oder Zeitimpulsgebers per Programm erfolgen, so steht dem Anwender die Routine EW.SUIZ zur Verfuegung. Dazu hat der Anwender die logische Geraete-Nr. in <A> einzutragen und die Routine zu rufen. Nach der Rueckkehr aus diesem Unterprogramm steht in <A> das startausloesende Datenwort und in <C> die zugehoerige Portadresse. Zur Ausloesung des Starts hat der Anwender den aus dem UP EW.SUIZ uebergebenen Wert <A> nach der in <C> uebergebenen Adresse auszugeben.
Am Ende des Zeitimpulses kann ein Interrupt generiert werden. Die Kanaele 2 und 3 arbeiten als vom Zeitgeber unabhaengige, kettbare Impulszaehler.

- Variante Zeitmesser

Fuer Zeitmessungen werden 3 oder 4 Kanaele gekettet (im ersten Fall wird Kanal 3 als unabhaengiger Impulszaehler mit einer Kapazitaet von 256 Bit verwendet). Dazu wird der Systemtakt um den Faktor 2 geteilt an den Eingang von Kanal 0 gelegt. Durch Freigabe bzw. Sperren dieser Impulse ueber den Torungseingang des Kanals (TE0) kann die Dauer eines Prozess-

signals ermittelt werden. Bei Kettung der Kanaele 0...2 betraegt die Maximalzeit 13,6s , bei Kettung der Kanaele 0...3 3496 s.

g) Standards:
Wird keine Adresse und kein Zeiger auf die Adresse eines Eingabepuffers im Ruf angegeben, so gilt die in der Zuweisungs-Tabelle fest definierte Adresse.
Wird keine Eingabemaske angegeben, so erfolgt die Eingabe unmaskiert.

h) Initialisierung:
Die im System EIEX 1521 vorgesehene Standard-Initialisierungsroutine kann als Unterprogramm in den Initialisierungsroutinen der logischen Geraete vom Typ UIZ aufgerufen werden, nachdem ihre Parameterversorgung sichergestellt wurde.
Die Initialisierung erfolgt unter Verwendung einer geraetespezifischen Initialisierungstabelle, deren Inhalt die Funktion des Geraetes eindeutig festlegt.
Die physische Adresse des Kanals 0 wird der Zuweisungs-Tabelle entnommen, alle anderen benoetigten Adressen werden daraus errechnet.
Mit Hilfe der Initialisierungsroutine werden die Initialisierungsmaske, PIO-Datenworte sowie Steuerworte und Zeitkonstanten der CTC-Kanaele aus der Initialisierungstabelle in die E/A-Tabelle kopiert. Ausserdem werden die kanalzugehoerigen Ueberlaufzaehler der E/A-Tabelle gleich Null gesetzt.

Parameter

<HL> = Adresse der Initialisierungstabelle

Aufbau der Initialisierungstabelle:

Byte-Nr.	Bedeutung
1	Interruptvektor CTC
2	Initialisierungsmaske
3	Datenwort PIO-Port A
4	Datenwort PIO-Port B
5	Steuerwort CTC-Kanal 0
6	Zeitkonstante fuer Kanal 0
7	Steuerwort CTC-Kanal 1
8	Zeitkonstante fuer Kanal 1
9	Steuerwort CTC-Kanal 2
10	Zeitkonstante fuer Kanal 2
11	Steuerwort CTC-Kanal 3
12	Zeitkonstante fuer Kanal 3
13	Kettenkennzeichnung

Aufbau der Initialisierungsmaske:

Bit	
Bit 0	Neuinitialisierung von Ports/Kanaelen
1	PIO-Port A
2	PIO-Port B
3	CTC-Kanal 0
4	CTC-Kanal 1
5	CTC-Kanal 2
6	CTC-Kanal 3
7	x

Fuer die Erlaeuterung der weiteren Elemente der Initialisierungstabellen werden folgende Abkuerzungen eingefuehrt:

ZSK Zaehlsignaleingang (Koppelbus)
ZAK Zeitimpulsgeber-Ausgang (Koppelbus)

Aufbau "Kettenkennzeichnung"

Bit 0	0
1	Kanal 0-1, 0-1-2, 0-1-2-3 gekettet
2	Kanal 2-3 gekettet
3	0
4	0
5	0
6	0
7	0

Bit 1, 2 werden nur bei Impulszaehler - und Zeitmesser - Anwendungen ausgewertet, sie sind sonst gleich Null zu setzen.

Bedeutung der Datenwort-Bits:

- Port A:

DB 0 = 1 ZSK wirkt als Torungssignal fuer Zaehleingang ZE 0
ZSK = H : ZE 0 freigeben

DB 1 = 1 Frequenzmesser; Freigabe von ZE 2 waehrend der Laufzeit des Zeitimpulsgebers

DB 2 = 1 Frequenzmesser 2 x 8 Bit; Freigabe von ZE 3 waehrend der Laufzeit des Zeitimpulsgebers

DB 3 = 1 macht Prozess-Signal an TE 0 unwirksam (A 3 = HIGH, wenn Torung von ZE 0 durch ZSK erfolgt)

DB 4 = 1 CTC-Kanal 1, 2 gekettet

DB 5 = 1 CTC-Kanal 2, 3 gekettet

DB 6 = 1	ZSK wirkt als Zaehlsignal fuer CTC-Kanal 0; (B 2 = H, B 3 = H notwendig) Zeitimpulsgeber- oder Frequenzmesserstart bei ZSK = H/L --> CLK/TRG 0 = H/L
DB 7 = 1	ZE 2 wirkt auf CLK/TRG 2 des CTC

- Port B:

DB 0 = 1	TE 2 freigeben (B0 = LOW: ZE 2 unabhaengig von TE staendig offen; Frequenzmesser)
DB 1 = 1	TE 3 freigeben (B 1 = LOW:ZE 3 unabhaengig von TE 3 staendig offen; Frequenzmesser 2 x 8 Bit)
DB 2 = 1	macht Prozess-Signal an ZE 0 unwirksam (notwendig in Verbindung mit A 6 = HIGH,B 4 = HIGH oder B 3 = H/L)
DB 3 = 1	ZE 0 freigegeben Port B DB 3 = 0/1 fuer Softwarestart des Zeitimpulsgebers bzw. ausloesung einer Frequenzmessung (B 2 = H notwendig)
DB 4 = 1	Zeitmesser; Freigabe des Systemtakts fuer CLK/TRG 0 des CTC (B 2 = H, B 3 = H notwendig)
DB 5 = 1	Kanal 0/1 gekettet
DB 6 = 1	ZE 3 wirkt auf CLK/TRG 3 des CTC
DB 7 = 1	ZE 1 wirkt auf CLK/TRG 1 des CTC

Programmierungstabelle der Vorzugsvarianten

Funktionsvariante	Datenwort (hexadezimal) Port A	Port B
Impulszaehler 4 x 8 Bit	80	CB
1 x 16 Bit (Kanaele 0, 1 gekettet) + 2 x 8 Bit	80	6B
2 x 16 Bit (Kanaele 0, 1 und 2, 3 gekettet)	A0	2B
1 x 24 Bit (Kanaele 0, 1, 2 + 1 x 8 Bit gekettet)	10	6B
1 x 32 Bit	30	2B

Frequenzmesser				
2 x 8 Bit (Start vom Prozess)	86			68
1 x 16 Bit (Start vom Prozess)	A2			2A
2 x 8 Bit (Start per Programm)	86	64	Start	6C
1 x 16 Bit (Start per Programm)	A2	26	Start	2E
Zeitimpulsgeber				
(Start vom Prozess) + Impulszaehler 2 x 8 Bit	80			6B
(Start vom Prozess) + Impulszaehler 1 x 16 Bit	A0			2B
(Start per Programm) + Impulszaehler 2 x 8 Bit	80	67	Start	6F
(Start per Programm) + Impulszaehler 1 x 16 Bit	A0	27	Start	2F
Zeitmesser				
2250 + Impulszaehler 1 x 8 Bit	10			7F
2330	30			3F

- Variante Impulszaehler

1. Interruptvektor entsprechend Speicherfestlegung zur Adressierung der Interruptserviceroutinen zu Kanal 0 ausgeben, falls Interruptabgabe eines oder mehrerer Kanaele bei Nulldurchgang des Zaehlerstandes vorgesehen ist. Die naechstfolgenden geraden Adressen sind fuer die Interruptserviceroutinen der Kanaele 1, 2, 3 zu reservieren, wenn diese Kanaele einen Interrupt generieren sollen.

2. Kanalsteuerwort
   Format: I 1 X S X 1 1 1
   - X beliebig
   - I = 0 Interrupt gesperrt
   - I = 1 Interrupt zugelassen
   - S = 0 Triggerflanke negativ (auf CTC-Zaehleingang bezogen)
   - S = 1 Triggerflanke positiv

mit X = 0 ergeben sich folgende Kanalsteuerworte

	S = 0	S = 1
I = 0	47H	57H
I = 1	C7H	D7H

3. Zeitkonstante TC

variabel, TC = 01H ... FFH = 1 ... 255

00H wird als 256 interpretiert

fuer gekettete Kanaele gilt

$$TC = TC_0 \times TC_1 \times TC_2 \times TC_3$$

Kanalsteuerworte und Zeitkonstanten werden nacheinander fuer alle aktiven Kanaele ausgegeben. Die Reihenfolge wird nach der Wertigkeit festgelegt (0, 1, 2, 3).

- Variante Zeitmesser

Kanaele 0 ... 3 werden in Betriebsart Zaehler programmiert (s.u. Impulszaehler).

1. Interruptvektor (falls erforderlich)
2. Kanalsteuerwort } fuer alle aktiven Kanaele
3. Zeitkonstante }

- Variante Zeitimpulsgeber

Kanal 0 wird in Betriebsart Zeitgeber, die Kanaele 1 ... 3 in Betriebsart Zaehler (s.u. Impulszaehler) programmiert.

1. Interruptvektor (falls erforderlich)
2. Kanalsteuerwort } fuer alle aktiven Kanaele
3. Zeitkonstante }

Kanal 0

Kanalsteuerwort 0 0 P 0 1 1 1 1

P = 0 Vorteiler = 16 Kanalsteuerwort 0FH

P = 1 Vorteiler = 256 Kanalsteuerwort 2FH
Zeitkonstante TC 0

TC_0 = 01H ... FFH = 1 ... 255

00H wird als 256 interpretiert

Zeitimpulsdauer t

$t_0 = T_0 \times P_0 \times TC_0 \times TC_1$

T_0 Periodendauer des Systemtaktes

Initialisierung der Kanaele 1 ... 3 (s.u. Impulszaehler).
Reihenfolge der Initialisierung: Kanal 0, 1, 2, 3
Vor jeder Zeitimpulsausgabe sind die Kanaele 0 und 1 neu zu initialisieren.

- Variante Frequenzmesser

Kanal 0 wird in Betriebsart Zeitgeber (s.u. Zeitimpulsgeber), die Kanaele 1 ... 3 in Betriebsart Zaehler (s.u. Impulszaehler) programmiert.

1. Interruptvektor (falls erforderlich)
2. Kanalsteuerwort
   fuer jeden aktiven Kanal
3. Zeitkonstante

Reihenfolge der Initialisierung: Kanal 0, 1, 2, 3
Vor jeder Frequenzmessung sind alle aktiven Kanaele neu zu initialisieren.

Bei der Festlegung oder Aenderung der Funktionsvariante des Moduls ist zuerst der PIO, dann der CTC zu programmieren.

3.8.4.10. Analoge Eingabe ueber die Prozess-Steckkarte AE-G AEG

a) Aufgabe:
Multiplexerfassung von bipolaren oder unipolaren analogen Signalen in folgenden, fuer das angesprochene logische Geraet bei der Generierung festgelegten Arten:
- Eingabe im schnellen ungeteilten Polling-Betrieb
- Eingabe im geteilten Verkehr

b) Status: Ruf

c) Schreibweise:

```
[name] PIN  R=n,(AEG=i[,ECA=(adr!symb)]
                    [,BOB=(adr!symb)]
                    [,((MASK=m!POM=(adr!symb)
                       !MUX=(adr!symb))]
                    [,SCV]
                    [,WAIT]
                 ! PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
AEG	–	Prozess-Steckkarte	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adresse des Fehlerbytes	
BOB	–	Adresse des Eingabepufferbereiches	
MASK	–	Eingabemasken-Byte	
POM	–	Zeiger auf Eingabemaske	
MUX	–	Zeiger auf Liste "Multiplexer-Adressen"	
SCV	–	Ablegen des Geraete-Kontrollwertes im Puffer	
WAIT	–	Aktivierung der WAIT-Funktion	
PARA	–	Adresse des indirekten Rufparameterblockes	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	

d) Ruf-Nummer: 7

e) Ruf-Laenge: 4 - 11 Byte

f) Wirkung:

Entsprechend der geraetemaessigen Ausstattung mit Erweiterungs-karten (AE-E) erfolgt die Analogeingabe von 8-56 Kanaelen im Multiplex-Betrieb.

Die Uebertragung in den in der Zuweisungs-Tabelle oder im Ruf definierten Eingabepuffer erfolgt in aufsteigender Reihenfolge entweder der Kanalnumerierung oder eines im Ruf vereinbarten Adressen-Verzeichnisses. Ist im Ruf kein Adressen-Verzeichnis angegeben, werden bei Angabe einer Eingabemaske die Kanaele nicht abgefragt, die in der zugeordneten Bitposition der Maske mit "1" maskiert sind. Dabei ist Bit 0 des ersten Maskenbytes dem Kanal 0 fest zugeordnet. Eine Maske kann entweder direkt durch die Angabe eines Maskenbytes oder indirekt durch die Angabe eines Zeigers auf ein Maskenfeld definiert werden. Ist die Maske direkt definiert, so werden nur die ersten 8 Kanaele maskiert eingegeben. Ist der Kanal 0 fuer die Geraeteueberwachung reserviert, wird fuer diesen Kanal eine Eingabemaske ignoriert. Bei einer maskierten Eingabe erfolgt die Uebertragung in einen lueckenlosen Pufferbereich. Ist eine Geraeteueberwachung initialisiert, kann bei Angabe der entsprechenden Option die Eingabe des Kontrollwertes von Kanal 0 in die erste Position des Puffers erfolgen.

Bei Angabe einer Adresse fuer das Fehlerbyte im Ruf koennen folgende Fehlercodes in den adressierten Speicher uebertragen werden:

0, 10H, 20H, 42H, 43H (siehe Anlage 3, Fehlerausschriften und Systemnachrichten EIEX 1521).

Die umgesetzten Werte besitzen das Format einer 2-Byte-Festkommazahl mit dem Wertebereich

-1 ... +0.999512 fuer bipolare Werte oder
0 ... +0.999756 fuer unipolare Werte.

- Variante AES1

Die Bedienung der Prozesskarte erfolgt im Interruptbetrieb.

- Variante AES2

Die Bedienung der Prozesskarte erfolgt im unterbrechbaren Pollingbetrieb.

g) Standards:
Wird keine Adresse des Eingabepuffers definiert, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse.
Ist das Geraet fuer eine Geraeteueberwachung initialisiert, so wird ein Fehlerbit im Status gesetzt, falls eine unerlaubte Abweichung bei der Umsetzung des auf dem Multiplexer-Kanal 0 aufgeschalteten Kontroll-Wertes auftritt. Der umgesetzte Kontrollwert wird bei Angabe der Option SCV in die erste Position des Eingabepuffers eingetragen.
Ist keine Eingabemaske oder Zeiger auf eine Eingabemaske im Ruf angegeben, so erfolgt die Eingabe unmaskiert.

h) Initialisierung:
Fuer die Initialisierung der Prozesskarte AE-G stehen die Standard-Routinen IAEG1 zur Verfuegung.
Als Parameter ist die Adresse einer Initialisierungstabelle in HL anzugeben, die folgenden Aufbau besitzen muss:

Byte-Nr.	Bedeutung	Bemerkungen
1	Interruptvektor	nur bei Interrupt-Betrieb
2	LSB des Pruefwertes	Bit 0="1":Geraetekontrolle Bit 1="1":Angabe des zul. Fehlers
3	MSB des Pruefwertes	nur fuer Byte 2:Bit 0 ="1"
4	zul. Fehler in 0/00	nur fuer Byte 2:Bit0+1="1"

Die Angabe des Pruefwertes erfolgt als positive Festkommazahl doppelter Genauigkeit, wobei die beiden niederwertigen Bits zur Steuerung der Initialisierung verwendet werden.
Die Angabe des zulaessigen Fehlers erfolgt in Promille in einem Byte als Integerzahl. Wird keine Angabe zum zul. Fehler gemacht, wird standardmaessig ein zulaessiger Fehler von 1 Prozent angenommen.
Die Einspeisung der Pruefspannung auf der Prozesskarte (siehe technische Beschreibung des Systems ursadat 5000).

3.8.4.11. Analoge Ausgabe ueber die Prozess-Steckkarte AA-1K

a) Aufgaben:
Ausgabe eines digitalen Wertes aus einem Ausgabepuffer auf einen Modul AA-1K zur Umsetzung in einen analogen Wert in folgenden durch die Belegung des ausgegebenen Datenwortes festgelegten Arten:
- Ausgabe eines Doppelbytes auf ein in der Zuweisungs-Tabelle adressiertes Port zur Umsetzung in einen unipolaren Wert - 12 Bit Datenbreite.
- Ausgabe eines Doppelbytes auf ein in der Zuweisungs-Tabelle adressiertes Port zur Umsetzung in einen bipolaren Wert - 11 Bit Datenbreite + Vorzeichen.

b) Status: Ruf

c) Schreibweise:

```
[name] POUT  R=n,(AA1K=i[,ECA=(adr!symb)]
                        [,BOB=(adr!symb)]
                        [,USER=(adr!symb)]
                    !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten:

R	-	Registerrettung	
n	-	Registerrettungsart	$0 \leq n \leq 2$
AA1K	-	Prozess-Steckkarte	
i	-	logische Geraetenummer	$0 < i < 256$
ECA	-	Adresse des Fehlerbytes	
BOB	-	Adresse des Ausgabepufferbereiches	
USER	-	Adresse einer anwenderspezifischen ISR	

PARA - Adresse des indirekten Rufparameterblockes
adr - absolute Adresse 0 < adr < FFFFH
symb - symbolische Adresse

d) Ruf-Nummer: 7
e) Ruf-Laenge: 4 -13 Byte
f) Wirkung:
Von einem in der Zuweisungs-Tabelle oder im Ruf angegebenen Puffer werden zwei Byte ueber ein in der Zuweisungs-Tabelle adressiertes logisches Geraet zur Umsetzung in einen analogen Wert ausgegeben.
Die Umsetzung erfolgt von einem 12 Bit-Datenwort entweder in einen unipolaren oder in einen bipolaren analogen Strom- oder Spannungswert.
Fuer die einzelnen Varianten ergeben sich folgende Zuordnungen:

Varianten unipolar

Wertigkeit	10V	5mA (an 100 Ohm)	20mA (an 100 Ohm)	1-5mA (an 100 Ohm)	4-20mA (an 100 Ohm)
0	0,0000V	0,00000V	0,00000V	0,10000V	0,40000V
2^0	0,0024V	0,00012V	0,00049V	0,10010V	0,40039V
2^1	0,0049V	0,00024V	0,00098V	0,10020V	0,40078V
2^2	0,0098V	0,00049V	0,00196V	0,10039V	0,40156V
2^3	0,0195V	0,00098V	0,00391V	0,10078V	0,40313V
2^4	0,0391V	0,00195V	0,00781V	0,10156V	0,40625V
2^5	0,0781V	0,00391V	0,01563V	0,10312V	0,41250V

2^{6}	0,1563V	0,00781V	0,03126V	0,10625V	0,42500V
2^{7}	0,3126V	0,01563V	0,06252V	0,11249V	0,45001V
2^{8}	0,6252V	0,03126V	0,12503V	0,12499V	0,50002V
2^{9}	1,2503V	0,06252V	0,25006V	0,14997V	0,60004V
2^{10}	2,5006V	0,12503V	0,50012V	0,19994V	0,80008V
2^{11}	5,0012V	0,25006V	1,00024V	0,29988V	1,20015V
4095	10,0000V	0,50000V	2,00000V	0,50000V	2,00000V

Varianten bipolar

Wertig-keit	10V	5mA (an 100 Ohm)
-2048	-10,0000V	0,50000V
-2^{10}	5,0000V	0,24996V
-2^{9}	2,5000V	0,12498V
-2^{8}	1,2500V	0,06249V
-2^{7}	-0,6250V	-0,03124V
-2^{6}	0,3125V	0,01562V
-2^{5}	0,1562V	0,00781V

-2^{4}	0,0781V	0,00391V
-2^{3}	0,0391V	0,00195V
-2^{2}	0,0195V	0,00098V
-2^{1}	0,0098V	0,00049V
-2^{0}	-0,0049V	-0,00024V
0	0,0000V	0,00000V
2^{0}	+0,0049V	+0,00024V
2^{1}	0,0098V	0,00049V
2^{2}	0,0195V	0,00098V
2^{3}	0,0391V	0,00195V
2^{4}	0,0781V	0,00391V
2^{5}	0,1562V	0,00781V
2^{6}	0,3125V	0,01562V
2^{7}	+0,6250V	+0,03124V
2^{8}	1,2500V	0,06249V
2^{9}	2,5000V	0,12498V
2^{10}	5,0000V	0,24996V
+2047	+9,9951V	0,49967V

Der Aufbau des auszugebenden Datenwortes ist dabei folgender:



High-Byte (Port A) — Low-Byte (Port B)

7 6 5 4 3 2 1 0 — 7 6 5 4 3 2 1 0

Low-Byte, Bit 7–0: untere 8 Bit des Datenwortes

High-Byte, Bit 3–0: obere 4 Bit des Datenwortes

High-Byte, Bit 4:
- ="0": Umschaltung auf externen Analogwert
- ="1": Analogwert kommt vom adressierten Modul

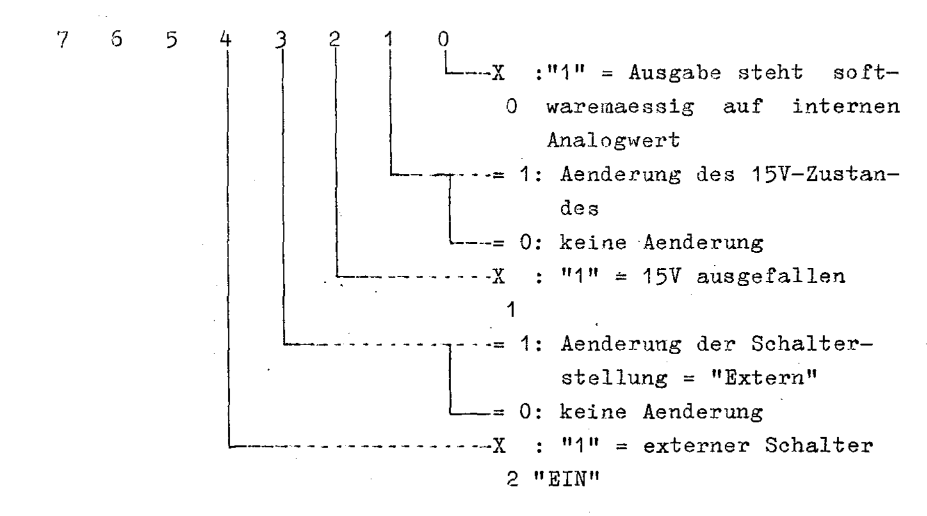
High-Byte, Bit 6: Kennung fuer Konvertierung
- ="0": Ausgabe eines bipolaren Wertes
- ="1": Ausgabe eines unipolaren Wertes

High-Byte, Bit 7: Kennung fuer Bit 4
- ="0": Bit 4 bleibt bei Ausgabe auf dem Modul unbeeinflusst
- ="1": Bit 4 wird vom Datenpuffer an den Modul ausgegeben

Durch das Abtesten des High-Bytes im Ausgabe-Datenpuffer erfolgt die Entscheidung ueber die Konvertierungsart (unipolar/bipolar) sowie ueber die Ausgabe eines externen oder internen Analogwertes.

Statusaenderungen der 15V-Versorgung oder Veraenderung der Schalterstellung - "Extern" werden ueber eine Interruptserviceroutine erfasst. Die Adressen zum Aktivierungsteil der Routinen sind entsprechend der Initialisierung in die Interruptvektortabelle einzubringen.

In diesem Aktivierungsteil wird die Registerrettungsroutine durchlaufen, die Adresse der zugehoerigen E/A-Tabelle in HL geladen und zur ISR verzweigt. Die ISR erfasst den Status des logischen Geraetes AA-1K, legt den aktuellen Zustand im Statusbyte der E/A-Tabelle ab und bereitet das Fehlerbyte nach folgendem Schema auf:



Wurde im letzten Ruf die Adresse eines Fehlerbytes angegeben, so wird dieses aufbereitete Byte dorthin ausgegeben.
Die im System EIEX 1521 enthaltene Standard-ISR kann durch einen anwenderspezifischen Teil ergaenzt werden. Es wird empfohlen, die Adresse von Fehlerbyte und anwenderspezifischen Teil der ISR stets anzugeben. Im anwenderspezifischen Teil der ISR kann das Fehler-Byte ausgewertet bzw. die Auswertung angemeldet werden (z.B. Fehlersignalisation).
Der Eingriff erfolgt im DI. Ueber Bit4 des Status-Byte kann im Anwenderteil der ISR Einfluss auf die Umschaltung "Extern" nach "Intern" genommen werden (z.B. bei stossfreier Umschaltung noetig), wobei die Adresse des Statusbyte in <HL> uebergeben wird.

g) **Standards:**
Wird keine Adresse des Ausgabepuffers definiert, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse.

h) Initialisierung:
Der PIO des logischen Geraetes wird Port B auf Byte-Ausgabe (Mode 0), Port A auf Bit-Ein-Ausgabe (Mode 3) initialisiert. Der Ausgang des logischen Geraetes wird definiert gesetzt auf interne Ausgabe des - je nach Bestueckungsvariante - Wertes 0 oder minus Maximalwert (-Max), bis durch einen Treiberaufruf ein neuer Analogwert erscheint. Ueber Register HL ist eine Tabelle INIT-PIOA folgender Form zu adressieren, womit der Interruptvektor in das Vektorregister geladen wird:

Byte-Nr.:	Bedeutung
1	Interruptvektor

3.8.4.12. Analoge Ausgabe ueber die Prozess-Steckkarte AA-5K

a) Aufgaben:
Ausgabe eines Bytes aus einem Ausgabepuffer auf einen Modul AA-5K zur Umsetzung in einen analogen Wert in folgenden bei der Generierung fuer das angesprochene logische Geraet festgelegten Arten:
- Ausgabe eines Bytes auf einen in der Zuweisungs-Tabelle definierten CTC-Kanal zur Umsetzung in einen unipolaren analogen Wert - 8 Bit Datenbreite (AA5KU)
- Ausgabe eines Bytes auf einen in der Zuweisungs-Tabelle definierten CTC-Kanal zur Umsetzung in einen bipolaren Wert 7 Bit Datenbreite + Vorzeichen (AA5KB).

b) Status: Ruf

c) Schreibweise:
```
[name] POUT  R=n,(AA5K=i[,ECA=(adr!symb)]
                         [,BOB=(adr!symb)]
                      !PARA=(adr!symb))[;Kommentar] .
```

Dabei bedeuten:

R	–	Registerrettung	
n	–	Registerrettungsart	$0 \leq n \leq 2$
AA5K	–	Prozess-Steckkarte	
i	–	logische Geraetenummer	$0 < i < 256$
ECA	–	Adresse des Fehlerbytes	
BOB	–	Adresse des Ausgabepufferbereiches	
PARA	–	Adresse des indirekten Rufparameterblockes	
adr	–	absolute Adresse	0 < adr < FFFFH
symb	–	symbolische Adresse	

d) Ruf-Nummer: 7

e) Ruf-Laenge: 4 - 11 Byte

f) Wirkung:

Von einem in der Zuweisungstabelle angegebenen Standardpuffer oder im Ruf angegebenen Datenpuffer wird ein Daten-Byte konvertiert und an den in der Zuweisungs-Tabelle adressierten Kanal ausgegeben.

Der Binaerwert hat die Wertebereiche von 0...250 fuer unipolare Analogwerte und von (0...125) fuer bipolare Analogwerte. Die von der Prozess-Steckkarte gelieferten analogen Signale sind von der Hardware-Konfigurierung der Karte abhaengig. Fuer die einzelnen Varianten ergeben sich folgende Zuordnungen:

Varianten unipolar

Wertigkeit	10V	5 mA (an 100 Ohm)
0	0V	0V
2^0	0,04V	0,002V
2^1	0,08V	0,004V

2^2	0,16V	0,008V
2^3	0,32V	0,016V
2^4	0,64V	0,032V
2^5	1,28V	0,064V
2^6	2,56V	0,128V
2^7	5,12V	0,256V
250	10,00V	0,500V

Varianten bipolar

Wertigkeit	10V	5mA (an 100 Ohm)
-125	-10,00V	-0,500V
-2^6	5,12V	0,256V
-2^5	2,56V	0,128V
-2^4	1,28V	0,064V
-2^3	0,64V	0,032V
-2^2	0,32V	0,016V
-2^1	0,16V	0,008V
-2^0	- 0,08V	-0,004V

0	0V	0V
2^0	+ 0,08V	+0,004V
2^1	0,16V	0,008V
2^2	0,32V	0,016V
2^3	0,04V	0,032V
2^4	1,28V	0,064V
2^5	2,56V	0,128V
2^6	5,12V	0,256V
+125	+10,00V	+0,500V

g) Standards:

Ist im Ruf keine Adresse des Ausgabepuffers angegeben, so gilt die in der Zuweisungs-Tabelle fest definierte Pufferadresse.

h) Initialisierung:

Zur Initialisierung der logischen Geraete ist die entsprechende Standard-Initialisierungsroutine (siehe Anlage 6, Anwendungsmoeglichkeiten der Prozess-Geraetetreiber) derart zu verwenden, dass die Adresse direkt in der Zuweisungs-Tabelle angegeben wird, da die Standard-Initialisierungsroutine keine zusaetzlichen Initialisierungsparameter benoetigt.

3.9. Dateiorganisation

3.9.1. Allgemeines

Der Anschluss der Folienspeicher MF 3200 erfolgt ueber die Anschlusseinheit AFS K 5121. Entsprechend der Generierung ist durch das Steuerprogramm EIEX 1521 eine Bedienung von maximal 8 Folienspeicherlaufwerken moeglich. Die einzelnen Laufwerke sind programmtechnisch ueber die Geraeteadressen von 1 bis 8 logisch adressierbar.

Es ist das Spurformat nach ISO TC 97/SC 11 Nr. 149 gewaehlt worden.

Spur 00 : Indexspur
Spuren 01 . . . 74 : Datenspuren
Spuren 75 und 76 : Austauschspuren

Jede Spur beinhaltet 26 Sektoren zu je 128 frei belegbaren Bytes. Die Numerierung der Sektoren laeuft von 01 bis 26.

Die Spur 00 des Folienspeichers wird als Indexspur bezeichnet. Sie ist fuer die normale Datenspeicherung fuer den Anwender gesperrt. Jeder Satz der Indexspur hat eine logische Satzlaenge von 80 Zeichen und eine physische Satzlaenge von 128 Zeichen. Die zwischen 81 und 128 liegenden Positionen enthalten Null(00). Die ersten sieben Sektoren der Indexspur enthalten Systeminformationen. Die verbleibenden 19 Sektoren (08 ... 26) enthalten Dateikennsaetze, die zur Definition der auf dem Folienspeicher stehenden Datensaetze dienen.

Sektor 07 : Datentraegerkennsatz (VOL)
Sektor 08 ... 26 : Dateikennsaetze (HDR)

Entsprechend dieser Festlegung ist je Folienspeicher die Aufzeichnung von maximal 19 Anwenderdateien moeglich.
Als Datei werden ein oder mehrere logische Saetze bezeichnet, die alle ueber einen gemeinsamen Namen ansprechbar sind. Diese Saetze koennen Programme oder auch nur Daten enthalten. Eine Datei auf dem Folienspeicher beginnt stets am Anfang eines Sektors. Die Groesse einer Datei ist nur durch den zur Verfuegung stehenden Platz auf dem Folienspeicher und durch die groesste in einem Ruf angebbare Satzanzahl (65535) bzw. Satzlaenge (3328) begrenzt. Innerhalb einer Datei ist jedoch nur eine Satzgroesse zulaessig. Eine Datei ueber mehrere Folienspeicher ist nicht moeglich. Als Aufzeichnungsverfahren wird das Satzformat geblockt und sequentiert verwendet (vgl. Robotron Standard K ROS-R 5108). Jeder Satz einer Datei wird mit Hilfe des Dateisteuerblocks (DSB) dieser Datei aufgerufen. Dieser wird bei Dateieroeffnung aufgebaut.

3.9.2. Kennsaetze

- Datentraegerkennsatz (VOL1)
  Der Datentraegerkennsatz dient der Identifikation des Datentraegers, des Benutzers, der physischen Satzfolge und der physischen Satzlaenge, den Zugriffsbedingungen und der Version des benutzten Standards.
  Der genaue Aufbau entspricht der Datentraeger-Richtlinie 8"-Diskette entsprechend dem Robotron-Standard (KROS-R 5108).

- Dateikennsatz (HDR1)
  Der Dateikennsatz dient der Identifikation der Datei und beschreibt ihre Lage auf dem Datentraeger. Er kennzeichnet die Verarbeitungsbedingungen der Datei.
  Der genaue Aufbau entspricht der Datentraeger-Richtlinie 8"-Diskette entsprechend des Robotron-Standard (KROS-R 5108).

3.9.3. Dateibedingungen

Eine Datei wird in einem physisch zusammenhaengenden Bereich auf dem Folienspeicher abgelegt. Fuer vorgesehene Erweiterungen ist ein entsprechender Bereich bei Dateivereinbarung festzulegen.

Die Dateiorganisation EIEX 1521 unterstuetzt die Zugriffsmethoden

- direkt adressiert
- sequentiell adresssiert

Die Zugriffsmethode ist unabhaengig von der Dateivereinbarung. Sie wird erst im Dateieroeffnungsruf spezifiziert.

- Direktzugriffsdateien
  Der direkte Zugriff erfolgt ausgehend von den im Zugriffsruf angegebenen Variablen.

  - Satznummer
  - Satzanzahl

  ueber die Umschluesselung der Satznummer in Verbindung mit den Angaben des internen Dateisteuerblocks relativ zum 1. Satz in Spur- und Sektornummer.

- Sequentielle Dateien
  Bei Eroeffnung einer sequentiellen Datei besteht ueber den Eroeffnungsruf die Moeglichkeit der Festlegung eines aktuellen Satzzaehlers oder der Uebernahme des Satzzaehlers vom HDR1-Satz der Datei in den internen DSB. Die sequentielle Erweiterung, Aktualisierung und Verarbeitung erfolgt ueber die im Zugriffsruf angegebene Variable Satzanzahl in Verbindung mit dem aktuellen Satzzaehler des internen DSB.

3.9.4. Interner Dateisteuerblock

Durch den Dateieroeffnungsruf OPEN wird fuer jede Datei ein interner Dateisteuerblock (DSB) aufgebaut.
Die maximale Anzahl der DSB ist bei der Generierung festzulegen.

Aufbau:

Byte-Nr.	Inhalt
1 bis 4	Dateiname 4 Zeichen im ISO-Code
5	Bit 5 = 1 POSF angewendet, sonst = 0 Bit 6 = 1 Schreibschutz gesetzt, sonst = 0 Bit 7 = 1 eroeffnet als Dateityp S 7 = 0 eroeffnet als Dateityp D
6	physische Laufwerknummer, auf der die Datei vereinbart ist
7	Sektornummer des Dateikennsatzes auf der Indexspur
8 9	Anfangsspur der Datei als duale Zahl Anfangsvektor der Datei als duale Zahl
10 - 11 12 - 13 14 - 15	Satzlaenge der Datei Satzanzahl der Datei Aktueller Satzzaehler der Datei

3.9.5. Rufe und Kommandos

3.9.5.1. Eroeffnung einer Datei OPEN

a) Aufgabe: - Aufbau des internen Dateisteuerblocks, ausgehend vom vorhandenen Dateikennsatz des angegebenen Dateinamens
- Festlegung des Dateityps
- Wahlweise Belegung des aktuellen Satzzaehlers der

Datei durch Angabe eines Anfangswertes bzw. Uebernahme des Satzzaehlers vom Dateikennsatz

- Festlegung von Schreibschutz

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] OPEN  R=n,(NAME=name,TYP=(S!D),FD=j[,FDS=z][,WRP]
                 ,ECA=(adr!symb)
                !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \le n \le 2$
NAME	- Dateiname	4 ISO-Zeichen
TYP	- Festlegung Dateityp	
S	- sequentielle Festlegung	
D	- direkte Festlegung	
ECA	- Fehlerschluesseladresse	
FD	- Floppy-Disk	
j	- logische Geraetenummer	$0 < j \le 8$
FDS	- Anfangswert des Satzzaehlers der Datei mit	$0 < z < 65535$
WRP	- Schreibschutz setzen	
PARA	- Parameteradresse	
adr	- absolute Adresse	$0 < adr < FFFFH$
symb	- symbolische Adresse	

d) Ruf-Nummer: 11

e) Ruflaenge: 12/14 Byte

f) Wirkung: Auf der Indexspur des Folienspeichers zum angegebenen Laufwerk erfolgt die Suche des zum angegebenen Dateinamen zugehoerigen Kennsatzes (HDR) und der Aufbau des internen Dateisteuerblocks (DSB) fuer diese Datei.

3.9.5.2. Schliessen einer Datei CLOS

a) Aufgabe: Loeschen des internen Dateisteuerblocks zum angegebenen Dateinamens und Aktualisierung des entsprechenden Dateikennsatzes.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] CLOS  R=n,(NAME=name,ECA=(adr!symb)
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
NAME	– Dateiname	4 ISO Zeichen
ECA	– Fehlerschluesseladresse	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nummer: 11

e) Ruflaenge: 10 Byte

f) Wirkung: Aktualisierung des zugehoerigen Dateikennsatzes bei Dateien vom Typ S durch Uebernahme des aktuellen Satzzaehlers des internen Dateisteuerblocks in den Dateikennsatz.
Loeschen des zugehoerigen internen Dateisteuerblocks

3.9.5.3. Positionieren einer Datei auf Anfang POSF

a) Aufgabe: Positionieren eroeffneter Dateien vom Typ S auf Dateianfang

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] POSF  R=n,(NAME=name,ECA=(adr!symb)
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
NAME	– Dateiname	4 ISO Zeichen
ECA	– Fehlerschluesseladresse	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nummer: 11

e) Ruf-Laenge: 10 Byte

f) Wirkung: Die Anwendung des Rufes ist nur auf eroeffnete Dateien vom TYP S moeglich. Es erfolgt im internen Dateisteuerblock eine Ueberschreibung des Wertes "Satzanzahl der Datei" mit dem Wert "Aktueller Satzzaehler" und Neufestlegung des Wertes "Aktueller Satzzaehler" auf eins.
Die erfolgte Positionierung wird im internen Dateisteuerblock gekennzeichnet.

3.9.5.4. Eroeffnen einer Dateierklaerung FIDE

a) Aufgabe: Aufbau des Dateikennsatzes

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] FIDE  R=n,(NAME=name,FD=j,LDS=l,ADS=a,ECA=(adr!symb)
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$

NAME - Dateiname 4 ISO-Zeichen
FD - Floppy-Disk
j - logische Geraetenummer $0 < j \le 8$
LDS - logische Satzlaenge mit $1 < l < 3328$ (3 1/4 K)
ADS - logische Satzanzahl mit $1 < a < 65535$
ECA - Fehlerschluesseladresse
PARA - Parameteradresse
adr - absolute Adresse $0 < adr <$ FFFFH
symb - symbolische Adresse

d) Ruf-Nummer: 11
e) Ruf-Laenge: 15 Byte
f) Wirkung: Aufbau des Dateikennsatzes und Ausgabe des Kennsatzes auf die Indexspur des angegebenen logischen Geraetes.

3.9.5.5. Loeschen einer Dateierklaerung CFID

a) Aufgabe: Loeschen des Dateikennsatzes
b) Status: Ruf
c) Schreibweise:
Ruf:

```
[name] CFID  R=n,(NAME=name,FD=j,ECA=(adr!symb)
                   !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:
R - Registerrettung
n - Registerrettungsart $0 \le n \le 2$
NAME - Dateiname 4 ISO-Zeichen
FD - Floppy-Disk
j - logische Geraetenummer $0 < j \le 8$
ECA - Fehlerschluesseladresse
PARA - Parameteradresse
adr - absolute Adresse $0 < adr <$ FFFFH
symb - symbolische Adresse

d) Ruf-Nummer: 11

e) Ruf-Laenge: 10 Byte

f) Wirkung: Logisches Loeschen des zugehoerigen Dateikennsatzes auf dem angegebenen Geraet.

3.9.5.6. Lesen/Schreiben Folienspeicher DISK

a) Aufgabe: Lesen/Schreiben von Datensaetzen von/in eroeffnete Anwenderdateien.

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] DISK R=n,(NAME=name,IN=k,TYP=(S!D),NDS=(s!symb),
                 ADS=(a!symb),BOB=(adr!symb),ECA=(adr!symb)
                  !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	- Registerrettung	
n	- Registerrettungsart	$0 \leq n \leq 2$
NAME	- Dateiname	4 ISO-Zeichen
IN	- Kommando	k=1 - Lesen k=2 - Schreiben
TYP	- Festlegung Dateityp	
S	- sequentielle Festlegung	
D	- direkte Festlegung	
NDS	- Satznummer mit	$0 < s < 65535$
ADS	- Satzanzahl mit	$0 < a < 32767$
BOB	- Pufferanfangsadresse	
ECA	- Fehlerschluesseladresse	
PARA	- Parameteradresse	
adr	- absolute Adresse	0 < adr < FFFFH
symb	- symbolische Adresse	

11/6447

d) Ruf-Nummer: 11

e) Ruf-Laenge: 15/17 Byte

f) Wirkung:

- Bei TYP=D: Lesen/Schreiben der angegebenen Satzanzahl beginnend von der angegebenen Satznummer der Datei mit dem angegebenen Namen von/auf den angegebenen Pufferbereich.
- Bei TYP=S: Lesen/Schreiben der angegebenen Satzanzahl beginnend vom Wert des aktuellen Satzzaehlers (aus internen Dateisteuerblock) der Datei mit dem angegebenen Namen von/ auf den angegebenen Pufferbereich.
  Fortschreibung des aktuellen Satzzaehlers im internen Dateisteuerblock um die angegebene Satzanzahl.
- Bei TYP=S und durchgefuehrter Positionierung POSF:
  Lesen und Fortschreibung aktueller Satzzaehler wie bei TYP=S
  Sonderfall:
  WERT AKTUELLER SATZZAEHLER + SATZANZAHL - 1 > VORHANDENE SATZANZAHL DATEI
  Beim Erreichen dieser Grenze werden nur noch die zulaessigen Datensaetze der Datei in den Pufferbereich uebertragen. Der Restpuffer wird mit dem Wert 00H belegt und die Fehlerzelle erhaelt den Wert 80H.

3.9.5.7. Datentraegerkennsatz schreiben INIT

a) Aufgabe: Aufbau des Datentraegerkennsatzes

b) Status: Task

c) Schreibweise:

Ruf:

```
[name] RUN R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]
```

d) Wirkung:

- Ueber ein bildschirmorientiertes Dialogeingabeprogramm erfolgt die Festlegung der variablen Daten des Datentraegerkennsatzes.
- Schreiben des Kennsatzes
- Die Task zum Aufbau des Datentraegerkennsatzes arbeitet immer mit dem logischen Laufwerk 1.

Bildschirmeingaben:

VOLUME IDENTIFIER: Eingabe ISO-Zeichen (max.6), als Name der Diskette

OWNER IDENTIFIER: Eingabe ISO-Zeichen (max.14), als Eigentuemeridentifikator

3.10. Spezielle Leistungen des EIEX

3.10.1. Eingabekonvertierung

3.10.1.1. Allgemeine Festlegungen

Der EIEX-Ruf ICON dient zur Konvertierung einer Zahl von der externen Darstellung im ISO-7-Bit-Code in ihre interne Darstellung.
Die Anzahl der in die Konvertierung einzubeziehenden Zeichen ist variabel aber fest vorzugeben.
Bei variabler Zeichenkettenlaenge kann der Bereich der unkonvertierten Zahl beliebig viele fuehrende Trennzeichen enthalten und muss durch mindestens ein Trennzeichen abgeschlossen sein. Als Trennzeichen sind alle Zeichen ausser +, -, . und Ziffer zugelassen.
Bei fester Zeichenkettenlaenge sind fuer den Bereich der unkonvertierten Zahl nur die Zeichen +, -, . und Ziffer zulaessig.
Treten Fehler in der Zeichenfolge einer Zahl auf, wird die Zahl nicht konvertiert. Der Fehler wird in der Fehlerzelle des Rufes registriert und es erfolgt die Rueckkehr in das aufrufende Programm.
Ist das im Ruf angegebene Datenformat nicht generiert, erfolgt eine Fehlermitteilung. Der zu konvertierende Zahlenbereich sowie die Fehlerzelle bleiben unveraendert.

- Interne Datenformate

. Integer einfache Genauigkeit - Format D2

Darstellung: 2 Byte, das hoechstwertige Byte steht auf der Anfangsadresse des Zielbereichs und sein Bit7 ist Vorzeichen-Bit; Wertigkeit faellt von links nach rechts.

Zahlenbereich: - 32768 < D2 < + 32767

. Integer doppelte Genauigkeit - Format D4

Darstellung: 4 Byte, das hoechstwertige Byte steht auf der Anfangsadresse des Zielbereichs und sein Bit7 ist Vorzeichen-Bit; Wertigkeit faellt von links nach rechts.

Zahlenbereich: - 2 147 483 647 < D4 < 2 157 483 647

. Festkomma einfache Genauigkeit - Format F2

Darstellung: 2 Byte, das hoechstwertige Byte steht auf der Anfangsadresse des Zielbereichs und sein Bit7 ist Vorzeichen-Bit; Wertigkeit faellt von links nach rechts.

Zahlenbereich: - 1.0000 < F2 < 0.9999
(4 signifikante Ziffern nach Punkt (.) nur zulaessig)

. Festkomma doppelte Genauigkeit - Format F4

Darstellung: 4 Byte,das hoechstwertige Byte steht auf der Anfangsadresse des Zielbereichs und sein Bit 7 ist Vorzeichen-Bit; Wertigkeit faellt von links nach rechts.

Zahlenbereich: - 1.000 000 000 < F4 < 0.999 999 999
(9 signifikante Ziffern nach Punkt (.) nur zulaessig)

- Konvertierungsprogramme

Folgende Konvertierungsprogramme sind generierbar:

. Integer : Realisiert die Formate D2 und D4
. Festkomma: Realisiert die Formate F2 und F4

3.10.1.2. Ruf Eingabekonvertierung ICON

a) Aufgabe: Eingabekonvertierung

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] ICON  R=n,(TYP=(D2!D4!F2!F4),SOB=(adr!symb),LOB=l,
                    DOB=(adr!symb),ECA=(adr!symb)
                   !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
TYP	– Datenformat	
D2	– Integer, einfache Genauigkeit	
D4	– Integer, doppelte Genauigkeit	
F2	– Festkomma, einfache Genauigkeit	
F4	– Festkomma, doppelte Genauigkeit	
LOB	– Zeichenkettenlaenge	variabel: $l = 0$ fest: $0 < l < 128$
SOB	– Anfangsadresse Quellbereich	
DOB	– Anfangsadresse Zielbereich	
ECA	– Fehlerschluesseladresse	
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < oder < FFFFH
symb	– symbolische Adresse	

f) Wirkung: Eingabekonvertierung der Zeichenkette vom Quellbereich entsprechend dem angegebenen internen Datenformat auf den Zielbereich.

g) Konvertierungsfehler: Fehlerschluessel

	Fehlerschluessel
. Zahlenbereichsueberschreitungen	80H
. Formatfehler	01H

3.10.2. Ausgabekonvertierung

3.10.2.1. Allgemeine Festlegungen

Der EIEX-Ruf OCON dient zur Konvertierung einer Zahl von der internen Darstellung in den externen ISO-7-Bit-Code.
Die Anzahl der bereitgestellten ISO-7-Bit-Zeichen ist abhaengig vom Format der internen Darstellung, jedoch konstant. Fuer Vornullen erfolgt die Bereitstellung des Zeichens "SPACE".
Ist das im Ruf angegebene Datenformat nicht generiert, erfolgt eine Fehlerermittlung.
Die interne Zahlendarstellung bleibt unveraendert.

- Interne Datenformate
  Festlegungen gemaess Eingabekonvertierung

- Konvertierungsprogramme
  Folgende Konvertierungsprogramme sind generierbar.
  . Integer : Realisiert die Formate D2 und D4.
  . Festkomma: Realisiert die Formate F2 und F4.

3.10.2.2. Ruf Ausgabekonvertierung OCON

a) Aufgabe: Ausgabekonvertierung
b) Status: Ruf
c) Schreibweise:
   Ruf:

```
[name] OCON  R=n,(TYP=(D2!D4!F2!F4),SOB=(adr!symb),
                  DOB=(adr!symb)
                 !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R   - Registerrettung
n   - Registerrettungsart $0 \leq n \leq 2$
TYP - Datenformat
D2   - Integer, einfache Genauigkeit
D4   - Integer, doppelte Genauigkeit
F2   - Festkomma, einfache Genauigkeit
F4   - Festkomma, doppelte Genauigkeit
SOB - Anfangsadresse Quellbereich
DOB - Anfangsadresse Zielbereich
PARA - Parameteradresse
adr - absolute Adresse   0 < adr < FFFFH
symb - symbolische Adresse

f) Wirkung: Ausgabekonvertierung von der internen Zahlendarstellung im angegebenen Datenformat in eine Zeichenkette im ISO-7-Bit-Code im Zielbereich.

g) Konvertierungsfehler:
keine

h) Externe Darstellung

. Integer einfache Genauigkeit - Format D2
Bereichsgroesse extern: 8 Byte
Darstellung: (SP)(SP)(VZ)ZZZZZ

. Integer doppelte Genauigkeit - Format D4
Bereichsgroesse extern: 13 Byte
Darstellung: (SP)(SP)(VZ)ZZZZZZZZZZ

. Festkomma einfache Genauigkeit - Format F2
Bereichsgroesse extern: 9 Byte
Darstellung: (SP)(SP)(VZ)0.ZZZZ oder
(SP)(SP)-1.0000

. Festkomma doppelte Genauigkeit - Format F4
Bereichsgroesse extern: 14 Byte
Darstellung: (SP)(SP)(VZ)0.ZZZZZZZZZ oder
(SP)(SP)-1.000000000

(SP) - Zeichen 'SPACE'

(VZ) - Vorzeichen + oder -

3.10.3. Codewandlung

3.10.3.1. Allgemeine Festlegung

Die Aufzeichnung auf Datentraegern fuer den externen Datentraegeraustausch erfolgt bei der Anwendung von EIEX 1521 standardmaessig im ISO-7-Bit-Code.
Besteht die Notwendigkeit der Ein- bzw. Ausgabe von Daten ueber Datentraeger mit einem abweichenden Code, so kann eine Codewandlung durchgefuehrt werden. Das Funktionsprinzip beruht darauf, dass ueber eine definierte Code-Tabelle der gewuenschte Zielcode durch zeichenweise Substitution im Datenfeld erzeugt wird. Dabei wird die Bitkombination eines jeden Zeichens des Datenfeldes als relative Adressangabe zur Code-Tabelle interpretiert, in der das Codezeichen abgespeichert ist. Das in der Code-Tabelle stehende Zeichen wird ausgelesen und auf den Speicherplatz des Datenfeldes transportiert.
Dieses Verfahren ermoeglicht die Wandlung des Dateninhaltes eines Datenfeldes in jeden beliebigen Code. Der Anwender legt den Inhalt der Code-Tabelle entsprechend seinem Anwendungsfall selbst fest, mit dem der EIEX-Ruf CODE arbeitet. Prinzipiell ist dafuer zu sorgen, dass die Anfangsadresse der Code-Tabelle modulo 256 vereinbart wird. Es koennen mehrere Code-Tabellen durch den Anwender vereinbart und waehrend der Bearbeitung benutzt werden.

3.10.3.2. EIEX-Ruf Codewandlung CODE

a) Aufgabe: Wandlung des Zeichencodes in einem Datenfeld nach vorgegebener Code-Tabelle

b) Status: Ruf

c) Schreibweise:

Ruf:

```
[name] CODE   R=n,(COTA=(adr!symb),BOB=(adr!symb),LOB=l,
                    !PARA=(adr!symb))[;Kommentar]
```

Dabei bedeuten die Schluesselworte und Parameter:

R	– Registerrettung	
n	– Registerrettungsart	$0 \leq n \leq 2$
COTA	– Anfangsadresse der Codetabelle	
BOB	– Anfangsadresse des zu codierenden Datenfeldes	
LOB	– Laenge des zu codierenden Datenfeldes mit	
l	– Feldlaenge	$0 < l < 65535$
PARA	– Parameteradresse	
adr	– absolute Adresse	0 < adr < FFFFH
symb	– symbolische Adresse	

d) Ruf-Nr. 18

e) Ruflaenge: 9 Byte

f) Wirkung: Innerhalb eines Datenfeldes werden die dort gespeicherten Zeichen entsprechend des in einer Codetabelle definierten Codes nach vorgegebener Laenge gewandelt. Wurde die Anfangsadresse der Code-Tabelle nicht modulo 256 vereinbart, so erfolgt eine Fehlerinformation und die den Ruf ausloesende Task wird fortgesetzt.

3.10.4. EIEX-Rufe mit indirekter Parameterangabe

EIEX 1521 ermoeglicht dem Anwender bei allen EIEX-Rufen die dazugehoerende Parameterfolge auf frei waehlbare RAM- oder PROM-Speicherbereiche innerhalb eines Applikationsprogrammsystems zu definieren. Bei der Abarbeitung dieser EIEX-Rufe werden ueber eine Verbindungsadresse die Parameter der Rufroutine uebermittelt.
Die Notation des EIEX-Rufes erfolgt mit der Angabe der Registerrettungsart und der symbolischen Adresse des Parameterblockes im Parameterfeld. Die Angabe der Parameterfolge wird in einem Pseudo-Ruf vorgenommen, dessen symbolische Adresse mit der im Parameterfeld des EIEX-Rufes stehenden Adresse identisch sein muss. Die allgemeine Schreibweise des Pseudo-Rufes lautet:

Namensfeld	Operationsfeld	Parameterfeld
[name]	ADR	RUF=er,par... [;kommentar]

Erlaeuterung:
er - symbolischer EIEX-Ruf
par - Parameterfolge

Beispiel: EIEX-Ruf RUN mit zyklischem und verzoegertem Programmstart bei indirekter Parameterangabe

```
        RUN  R=n,PARA=BLOCK

BLOCK:  ADR  RUF=RUN,TASK=p,DET=T(HR!MI!SE!CL),
             CT=t(HR!MI!SE!CL)
```

4. Angaben zur Generierung des Steuerprogramms EIEX 1521

4.1. Allgemeine Hinweise

Die Generierung einer anwendungsspezifischen Version von EIEX 1521 erfolgt auf der Basis der Quellprogramme der System - Komponenten des EIEX 1521 und der evtl. Einbindung von anwendereigenen Programmoduln. Die Erzeugung von anwendungsspezifischen Echtzeitsteuerprogrammversionen kann somit in zwei Stufen unterteilt werden:

1. Generierung einer anwenderspezifischen Echtzeitsteuerprogrammversion aus den standardmaessig bereitgestellten Programmoduln des EIEX 1521 durch Selektion und Parametermodifikation. Im Ergebnis liegt ein arbeitsfaehiges Echtzeitsteuerprogrammsystem vor, das als Basis des Applikationsprogrammsystems dient.

2. Generierung eines anwendungsspezifischen Echtzeitsteuerprogrammsystems entsprechend dem o.g. Verfahren und Komplettierung der Programmversion um eigenstaendig entwickelte Programmoduln, die zusammen ein spezifisches Echtzeitsteuerprogrammsystem fuer ein Applikationsprogrammsystem ergeben.

Die geraetetechnische Voraussetzung fuer die Durchfuehrung der Generierung bildet das Mikrorechnerentwicklungssystem MRES A 5601 und das dazugehoerige Betriebssystem MEOS 1521. In Abhaengigkeit von der verfuegbaren Geraetetechnik kann die Generierung manuell oder maschinell durchgefuehrt werden, wobei die manuelle Generierung stets moeglich ist. Eine maschinelle Generierung, bei der durch ein dialogorientiertes Generierungsprogramm der Zusammenbau einer anwender-

spezifischen Echtzeitsteuerprogrammversion weitestgehend automatisch erfolgt, setzt jedoch als geraetetechnische Basis den MRES A 5601.20 und das Betriebssystem EMOS 1521 voraus. Bei einer Generierung werden durch das Generierungsprogramm die Quellprogramme der Systemkomponenten von EIEX 1521 ausgewaehlt und modifiziert.
Anschliessend erfolgt deren weitere Bearbeitung durch den Assembler und Linker, wobei die erzeugten Echtzeitsteuerprogrammversionen auf die Datentraeger Folienspeicher oder Lochband bzw. auf PROM's ausgegeben werden.

In Vorbereitung der Generierung von EIEX 1521 sind durch den Anwender entsprechend seinem anwendungsspezifischen Einsatzfall konzeptionelle Vorarbeiten zu leisten. Sie beinhalten bezueglich EIEX 1521 die Auswahl der fuer seinen Einsatzfall relevanten Programmoduln und die Festlegung der zu generierenden Parameter fuer die ausgewaehlten Programmkomponenten. Die Wechselbeziehungen zwischen den Bausteinen, die Aufrufbeziehungen der Parameter sowie die dazugehoerigen Daten sind als wesentliche Informationen aus den Beschreibungslisten zu entnehmen und in Check- Listen zu uebertragen. Diese Informationen bilden stets die verbindlichen Grundlagen fuer die unmittelbare Durchfuehrung der Generierung einer anwendungsspezifischen Echtzeitsteuerprogrammversion des EIEX 1521 (siehe Anleitung fuer den Systemprogrammierer zum Echtzeitsteuerprogrammsystem EIEX 1521).

4.2. Generierung der Komponenten des EIEX 1521

4.2.1. Konfigurationsbezogene Generierung

4.2.1.1. DV- Peripherie

Der Anwender hat fuer das zu generierende System die Anzahl der vorhandenen DV- Geraete anzugeben.

Die Zuordnung zwischen logischer Geraetenummer und physischem Geraet ist in Form einer Geraetezuweisungstab. vorzugeben. Der Aufbau der Geraetezuweisungstabelle ist unter Pkt. 4.3.7.2. an der Anleitung fuer den Systemprogrammierer beschrieben.

- Der Anwender kann alle Geraete der DV- Peripherie wahlweise mit bzw. ohne Warteschlange betreiben.
- Die DV- Geraete koennen wahlweise zeitlich ueberwacht werden, ihre Anzahl ist anzugeben.
  Alle Geraete sind in die Geraetezuweisungstabelle lueckenlos einzuordnen.
- Das Umschalten bzw. Rueckschalten von DV- Geraeten kann wahlweise erfolgen, das Gleiche gilt fuer die Verwendung des Rufes "WAIT". Diese Programmteile sind bei der Generierung einzubinden.

4.2.1.2. Prozess- Peripherie

Der Anwender hat fuer das zu generierende System eine Auswahl der gewuenschten Treiberroutinen vorzunehmen und die Anzahl der logischen Geraete anzugeben.
In Abhaengigkeit der gewaehlten Treiberroutinen ist die Zeitsteuerung fuer die Prozessperipherie zu generieren und die Anzahl der logischen Geraete anzugeben, die unter Zeitsteuerung arbeiten sollen.
Entsprechend der gewaehlten logischen Geraetenummern ist eine Zuordnung zum physischen Geraet in eine Geraetezuweisungstabelle vorzunehmen, sowie eine Einordnung der entsprechenden Interruptserviceroutinen in die Interrupt- Vektortabelle des EIEX vorzunehmen.

4.2.2. Anwendungsbezogene Generierung

4.2.2.1. Vorrangorganisation

- Angabe der Anzahl der Task im Applikationsprogrammsystem
- Die Task 1 und 2 koennen durch Systemprogramme belegt werden:

  Task 1 : Kommandoorganisation
  Task 2 : Systemnachrichtenorganisation

- Ueber die in der Vorrangorganisation immer enthaltenen Rufe bzw. Kommandos RUN und BYE sind alle weiteren wahlweise anwenderspezifisch zu generieren.

4.2.2.2. Interruptorganisation

- Festlegung einer Registerrettungsart (1 oder 2), die fuer Hardware- und Softwareinterrupt gueltig ist.
- Angaben der groessten zulaessigen Rufnummer der generierten EIEX- Rufroutinen
- Einordnung der symbolischen Adressen der im System benutzten Interruptservice- Routinen in die Interruptvektoradress -Tabelle.
- Einordnung der symbolischen Adressen der im System benutzten Rufroutinen in die Rufadresstabelle. Entfaellt eine Rufroutine, so wird standardmaessig die symbolische Adresse einer Fehlermeldungsroutine generiert.

4.2.2.3. Anlauforganisation

- Angabe der Task, die beim Systemstart anwendungsspezifische Anlaufteile enthaelt und im System als erste Task aktiviert wird.
  Entfaellt deren Angabe, so geht EIEX 1521 nach dem standardisierten Systemanlauf in den dynamischen Stop.

4.2.2.4. Echtzeituhr und Zeitorganisation

- Wahlweise Generierung des Kalenderprogrammes
- Festlegung der kleinsten Zeitbasis der Zeitorganisation
  Folgende Zeiteinheiten sind standardmaessig generierbar:

  10 ms, 20 ms und 25 ms

Mit der Wahl einer zusaetzlichen Programmkomponente koennen folgende Zeiteinheiten generiert werden:

50 ms, 100 ms, 200 ms, 250ms, 500 ms und 1000 ms.

- Maximale Anzahl der von der Zeitorganisation zu verwaltenden Tasks
  Stimmen die Zeitbereiche von Zyklus- und Verzoegerungszeit einer Task nicht ueberein, so wird fuer die Task in der Zeittabelle der doppelte Speicherplatz benoetigt. Demzufolge sind solche Tasks bei der Angabe der maximalen Anzahl als 2 Tasks zu beruecksichtigen.
- Wahlweise Generierung der Rufe / bzw. Kommandos der Zeitorganisation

4.2.2.5. Bedienerkommunikation

- Wahlweise Generierung der gesamten Bedienerkommunikation in den Komponenten:

  - Kommandoorganisation
  - Systemnachrichtenorganisation

- Wahlweise Zuordnung von MON 1 und/oder Drucker als Kommunikationsgeraet.
- Wahlweise Zuordnung von MON 1 und/oder Drucker zur Fehlerprotokollierung.

4.2.2.6. Unterprogrammorganisation

- Wahlweise Generierung der Komponente UP- Organisation
- Angabe der Anzahl der in der UP- Organisation zu verwaltenden Programme.

4.2.2.7. Dateiorganisation

- Wahlweise Generierung der Komponente Dateiorganisation
- Angabe der Anzahl der Dateien die durch EIEX 1521 gleichzeitig in eroeffnetem Zustand verwaltet werden sollen.

4.2.2.8. Spezielle Leistungen des EIEX

- Wahlweise Generierung der Komponenten

    - Eingabekonvertierung
    - Ausgabekonvertierung

  sowie wahlweise Generierung der zugehoerigen Zahlenformate

    - Integer
    - Festkomma

- Wahlweise Generierung des EIEX-Rufes Codewandlung

5. Aufwandsangaben

5.1. Speicheraufwand des EIEX 1521 in Byte

Saemtliche Angaben beziehen sich auf das generierte Maximum der entsprechenden Komponenten

5.1.1. Vorrangorganisation	PROM/BYTE	RAM/BYTE
- Vorrangentschluesselung	255	-
- Arbeitsbereiche konstant	-	5
- Arbeitsbereiche variabel		
. Zahl der Task (ZT)	4 x ZT	8 x ZT
- Rufe / Kommandos	-	1/8 x ZT
. BYE	44	-
. CNCL	96	20

. GO	105	–
. DISP	45	–
. ENAP	17	–
. RUN	42	–
. PAUS	43	–
. HELP	289	2
. CHAN	492	6
– Arbeitsbereich variabel		
. Zahl der Wechseltask (WT) bei CHAN	–	2 x WT+4xZT
– Zentrale Routinen	189	3

5.1.2. Interruptorganisation	PROM/BYTE	RAM/BYTE
– Hardwareinterruptsteuerung		
. mit Registerrettung 1	120	–
. mit Registerrettung 2	133	–
– Softwareinterruptsteuerung		
. mit Registerrettung 1	146	–
. mit Registerrettung 2	167	–
– Arbeitsbereiche konstant	–	57
– Arbeitsbereiche variabel		
. Zahl der ISR (n)	n x 2	n x 30
. Zahl der Rufe (m)	m x 2	–
. Systemstack je nach Generierung		x

5.1.3. Organisation des Systemanlaufes	PROM/BYTE	RAM/BYTE
– standardisierter Teil	235	–
– variabler Teil	158	–

5.1.4. Echtzeituhr- und Zeitorganisation

	PROM/BYTE	RAM/BYTE
– Uhrprogramm	67	4
– Kalenderprogramm	81	4
– Aktualisieren der Zeittabellen	239	10
– Arbeitsbereiche variabel		
. Zahl der Task (ZT)	–	5 x ZT
– Rufe / Kommandos		
. Run	243	–
. PAUS	31	–
. CNCL	24	–
. TIME	36	–
. DATE	57	–

5.1.5. Organisation der Unterprogramme

	PROM/BYTE	RAM/BYTE
– Arbeitsbereich konstant	135	–
– Arbeitsbereich variabel		
. Anzahl der Unterprogramme (ZU)	–	2 x ZU

5.1.6. DV - Organisation

	PROM/BYTE	RAM/BYTE
– Rahmensteuerung mit integrierter Rufbearbeitung fuer COTR, READ und WRIT	695	2
– Warteschlangenorganisation	153	–
– Zeitueberwachung mit Geraetefehler-behandlung	262	14
– Automatische Geraeteumschaltung	730	2
– Rufe		
. ASGN	595	3
. WAIT	111	

- Arbeitsbereich variabel		
. Zahl der DV - Geraete im Anwender-system (a)	a x 21	a x 34
. Zahl der zeitlich zu ueberwachenden Geraete (b)	-	b x 2
. Zahl der Eintragungen in die Warteschlangen (ZT)	-	a x ZT/8
- DV - Peripherie		
. Seriendrucker	1778	9
. Lochbandleser	448	-
. Lochbandstanzer	496	-
. Folienspeicher	2261	202
. Monitor	270	a x 2
. Tastatur	262	-

5.1.7. Bedienerkommunikation	PROM/BYTE	RAM/BYTE
- Tastatureingabe	485	20
- Kommandopruefung		
. GO	30	-
. LOG	8	-
. NOLOG	6	-
. RUN	382	-
. CNCL	100	-
. ENAP	32	-
. DISP	22	-
. ASGN	118	-
. SDATE	74	-
. STIME	76	-
. HELP	31	-
. CHAN	102	-
- Kommandoanzeige/Druck	116	-
- Fehlermeldung	152	-
- Zentrale Routinen	273	12

5.1.8. Systemnachrichtenorganisation

	PROM/BYTE	RAM/BYTE
- Zentrale Routinen	483	148
- Fehlerwortpuffer	876	-
- Arbeitsbereich variabel		
. Zahl der Eintragungen fuer den Puffer (Z)	-	Z x 10

5.1.9. Konvertierung

	PROM/BYTE	RAM/BYTE
- Ruf Eingabekonvertierung	51	-
- Ruf Ausgabekonvertierung	49	-
- Konvertierungsmodule		
. Eingabekonvertierung Integer	228	4
. Ausgabekonvertierung Integer	230	5
. Eingabekonvertierung Festkomma	465	24
. Ausgabekonvertierung Festkomma	165	8
- Ruf Codewandlung	52	-

5.1.10. File-Handler

	PROM/BYTE	RAM/BYTE
- Zentrale Routinen	998	79
- Systempuffer	-	128
└ pro eroeffnete Datei im internen Dateisteuerblock	-	15
- Zentraler Kern des File-Handlers	174	-
- Ruf OPEN	168	-
- Ruf GLOS	160	-
- Ruf FIDE	290	-
- Ruf CFID	111	-
- Ruf DISK	496	-
- Ruf POSF	74	-

5.1.11. PO - Organisation

	PROM/BYTE	RAM/BYTE
- Rahmensteuerung	462	7
- Initialisierung Prozessperipherie	103	-
- Ruf WAIT	109	-
- Arbeitsbereich variabel		
. Zahl der log. Geraete unter Zeitkontrolle (LZ)	-	3 x LZ

5.2. Rechenzeiten in Millisekunden

- Unterbrechung einer laufenden Task, Registerrettungsart 1 und Start einer Task hoeherer Prioritaet (entspricht Wirkung des EIEX- Rufes RUN) 0,760 ms

- Fortsetzen einer unterbrochenen Task nach Hardware- Interrupt (Registerrettungsart 1) 0,140 ms

- Maximale Durchlaufzeit durch die Vorrangorganisation

  (64 Tasks bei Systemstart und keine Taskanmeldung liegt vor) 0,340 ms

- Unterbrechung einer laufenden Task bei Durchlauf des Uhrprogrammes und Start eier Task hoeherer Prioritaet (es wird nur eine Task zyklisch in Grundtakt gestartet) 0,560 ms

- Einleiten einer Uebertragung auf ein nicht besetztes DV-Geraet mit Registerrettungsart1, Ruferoeffnung und Ruecksprung mit Registergenerierung in unterbrochene Task sowie deren Fortsetzung (ohne Bearbeitungszeit des Treibers). bers). 0,860 ms

- Unterbrechung einer Task durch die Uhr, Aktualisieren der Uhr (bei Stundendurchlauf) und Fortsetzen der Task
  0,370 ms

- Unterbrechung einer Task durch die Uhr, Aktualisieren der Uhr und des Kalenders bei Jahreswechsel und Fortsetzung einer Task
  0,490 ms

- Ein- u. Austritt fuer ein UP der zentralen UP- Bibliothek (UP ist nicht besetzt)
  0,180 ms

Anlage 1 : Uebersicht der EIEX- Rufe

Jeder EIEX- Ruf wird durch folgende 4 Punkte erlaeutert:

*** Rufname
1. Bedeutung
2. Rufaufbau
3. Rufnotation

In der Rufnotation bedeuten

```
( )   - Auswahl eines durch ! getrennten Elementes
[ ]   - Wahlweise
 !    - Oder- Zeichen
```

*** ASGN

1. Geraetezuweisung bzw. -umschaltung

```
2.  RST  n
    DB   9    ;  Rufnummer
    DB   l    ;  Parameterlaenge (l=2 ohne Fehlerschluessel
                                  l=4 mit  Fehlerschluessel)
    DB   i    ;  Logische Geraetenummer Defektgeraet
    DB   i    ;  Logische Geraetenummer Ersatzgeraet
    DA   adr  ;  Fehlerschluesseladresse
```

```
3.  ASGN   R=n,(SUD=i,SEQ=i[,ECA=(adr!symb)]
                !PARA=(adr!symb))[;Kommentar]
```

*** BYE

1. Beenden einer Task

```
2.      RST   n
        DB    1    ;    Rufnummer
        DB    m    ;    Parametermaske

        Parametermaske:
        DB    0    ;    BYE ohne Wiederstart
        DB    80H  ;    BYE mit Wiederstart

3.      BYE   R=n[,REP!,PARA=(adr!symb)][;Kommentar]
```

*** CHAN

1.1. Prioritaetswechsel zweier Task ohneRuecktragung

```
1.2.    RST   n
        DB    17   ;    Rufnummer
        DB    82H  ;    Parameterlaenge
        DB    p    ;    Tasknummer
        DB    p    ;    Wechseltasknummer

1.3.    CHAN       R=n,(TASK=p,CTASK=p!PARA=(adr!symb))[;Kommentar]
```

2.1. Prioritaetswechsel zweier Task mit Ruecktragung

```
2.2.    RST   n
        DB    17   ;    Rufnummer
        DB    2    ;    Parameterlaenge
        DB    p    ;    Tasknummer
        DB    p    ;    Wechseltasknummer
```

2.3. CHAN R=n,(TASK=p,CTASK=p,BYE!PARA=(adr!symb))[;Kommentar]

3.1. Aufheben des Prioritaetswechsels

```
3.2. RST  n
     DB   17   ;  Rufnummer
     DB   41H  ;  Parameterlaenge
     DB   p    ;  Taskrummer
```

3.3. CHAN R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

*** CNCL

1.1. Abbruch einer Task

```
1.2. RST  n
     DB   2    ;  Rufnummer
     DB   81H  ;  Parameterlaenge
     DB   p    ;  Tasknummer
```

1.3. CNCL R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

2.1. Austragen einer Task aus der zeitlichen Verwaltung

```
2.2. RST  n
     DB   2    ;  Rufnummer
     DB   1    ;  Parameterlaenge
     DB   p    ;  Tasknummer
```

2.3. CNCL R=n,(TASK=p,TM!PARA=(adr!symb))[;Kommentar]

3.1. Austragen aller zeitlichen Aktivierungen

```
3.2. RST    n
     DB     2    ;    Rufnummer
     DB     0    ;    Parameterlaenge

3.3. CNCL     R=n,(ALL!PARA=(adr!symb))[;Kommentar]
```

*** COTR

1.1. Loeschen des Nutzerbereiches des MON 1

```
1.2. RST    n
     DB     8    ;    Rufnummer
     DB     l    ;    Parameterlaenge  ( l=3 ohne, l=5 mit  Fehler-
                                                schluesseladresse)
     DB     m    ;    Parametermaske
     DB     i    ;    Logische Geraetenummer MON
     DB     1    ;    Maschinenkommando
                      Nutzbereich loeschen (k=1)
     DB    adr   ;    Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen.

Bit 2 : Fehlerschluesseladresse

```
1.3. COTR     R=n,(MON=i,IN=k,ECA=(symb!adr)
                   !PARA=(adr!symb))[;Kommentar]
```

2.1. Teilbereichsloeschen im Nutzerbereich des MON 1

```
2.2. RST    n
     DB     8    ;    Rufnummer
     DB     l    ;    Parameterlaenge  (l=9  ohne  Fehlerschluessel
                                        l=11  mit Fehlerschluessel)
```

```
DB   m   ;   Parametermaske
DB   i   ;   Logische Geraetenummer MON 1
DB   mk  ;   Maschinenkommando
             mk=05H: Teilloeschen mit Angabe der letzten
                     zu loeschenden Position (k=2)
             mk=0DH: Teilloeschen mit Angabe der Loesch-
                     laenge
DA   adr ;   Pufferadresse (adr=0)
DA  (zs!l);  Loeschposition, Loeschlaenge
             mk=05H: Loeschposition in Spalte (s) und Zei-
                     le (z)
             mk=0DH: Anzahl der zu loeschenden Zeichen
                     (< 1024)
DA   zs  ;   Loeschanfang in Spalte (s) und Zeile (z)
DA   adr ;   Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen.

Bit 0 : Loeschlaenge oder Loeschendeposition

Bit 1 : Loeschbeginnposition

Bit 2 : Fehlerschluesseladresse

2.3. COTR R=n,(MON=i,IN=k,BPOS=z/s,(EPOS=z/s!LOB=l), ECA=(adr!symb)!PARA=(adr!symb))[;Kommentar]

3.1. COTR fuer Seriendrucker SD 1156

```
3.2. RST  n
     DB   8   ;   Rufnummer
     DB   7   ;   Parameterlaenge
     DB   m   ;   Parametermaske
     DB   i   ;   Logische Geraetenummer
     DB   04H ;   Maschinenkommando
     DA   po  ;   Positioniergroesse (siehe WRIT fuer SD 1156)
     DA   adr ;   Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

Bit 1 : Angabe der Positioniergroesse
Bit 2 : Fehlerschluesseladresse
Bit 4 : Ruf mit WAIT

3.3. COTR R=n,(LP=i(,BPOS=d!(FLF=(z!s),LEP=(1!2!3)))[,WAIT] [,ECA=(adr!symb)]!PARA=(adr!symb))[;Kommentar]

4.1. COTR fuer Lochbandstanzer 1215

4.2.
```
RST   n
DB    8    ;   Rufnummer
DB    5    ;   Parameterlaenge
DB    m    ;   Parametermaske
DB    i    ;   Logische Geraetenummer
DB    mk   ;   Maschinenkommando
               mk = 20H:  LB-Austrieb  (k=1)
               mk = 40H:  LB-Rueckschritt  (k=2)
DA    adr  ;   Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

Bit 2 : Fehlerschluesseladresse
Bit 4 : Ruf mit WAIT

4.3. COTR R=n,(PTP=i,IN=k[,ECA=(adr!symb)][,WAIT]

!PARA=(adr!symb))[;Kommentar]

*** CODE

1. Codeumwandlung

```
2.    RST   n
      DB   18   ;   Rufnummer
      DB    6   ;   Parameterlaenge
      DA   adr  ;   Anfangsadresse der Codetabelle
      DA   adr  ;   Anfangsadresse des zu codierenden Bereiches
      DA    l   ;   Laenge des zu codierenden Bereiches

3. CODE   R=n,(COTA=(adr!symb),BOB=(adr!symb),LOB=l
                !PARA=(adr!symb))[;Kommentar]
```

*** DATE

1. Uebernahme des Datums

```
2.    RST   n
      DB   13   ;   Rufnummer
      DB    2   ;   Parameterlaenge
      DA   adr  ;   Zieladresse

3.    DATE   R=n,(BOB=(symb!adr)!PARA=(adr!symb))[;Kommentar]
```

*** DISP

1. Verhindern der Bearbeitung einer Task

```
2.    RST   n
      DB    5   ;   Rufnummer
      DB    1   ;   Parameterlaenge
      DB    p   ;   Tasknummer

3.    DISP   R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]
```

*** ENAP

1. Erlauben der Bearbeitung einer verhinderten Task

```
2.  RST   n
    DB    6   ;   Rufnummer
    DB    1   ;   Parameterlaenge
    DB    p   ;   Tasknummer
```

3. ENAP R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

*** GO

1. Fortsetzen einer pausierenden Task

```
2.  RST   n
    DB    4   ;   Rufnummer
    DB    1   ;   Parameterlaenge
    DB    p   ;   Tasknummer
```

3. R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

*** HELP

1. Prioritaetswechsel einer Task

```
2.  RST   n
    DB   16   ;   Rufnummer
    DB    1   ;   Parameterlaenge
    DB    p   ;   Tasknummer
```

3. HELP R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]

*** ICON

1. Eingabekonvertierung

2.
```
RST  n
DB   14   ;   Rufnummer
DB    8   ;   Parameterlaenge
DB    f   ;   Format
DB    l   ;   Laenge
DA   adr  ;   Quelladresse
DA   adr  ;   Zieladresse
DA   adr  ;   Fehlerschluesseladresse
```

Format: D2 - Integer 2 Byte
D4 - Integer 4 Byte
F2 - Festkomma 2 Byte
F4 - Festkomma 4 Byte

Laenge: l=0 - Konvertierung von Quelladresse bei abschliessendem Trennzeichen
l=n - Konvertierung von n Zeichen

3.
```
ICON    R=n,(TYP=(D2!D4!F2!F4),SOB=(adr!symb),LOB=l,
             DOB=(adr!symb),ECA=(adr!symb)
             !PARA=(adr!symb))[;Kommentar]
```

*** LISD

1. Aufruf eines nicht unterbrechbaren UP's aus der UP- Bibliothek

2.
```
DI
CALL    SR.EP    ;    ( SR - Symbolischer Name fuer UP
                        EP - Eintrittspunkt )
```

3. LISD LIB=symb[;Kommentar]

*** LISE

1. Aufruf eines unterbrechbaren UP's aus der UP- Bibliothek

2.
```
CALL   E4.SR   ;   SR - Symbolischer Name fuer UP
CALL   SR.EP   ;   EP - Eintrittspunkt
RST     28H
```

3. LISE LIB=symb[;Kommentar]

*** OCON

1. Ausgabekonvertierung

2.
```
RST   n
DB   15   ;   Rufnummer
DB    5   ;   Parameterlaenge
DB    f   ;   Format
DA   adr  ;   Quelladresse
DA   adr  ;   Zieladresse
```

3. OCON R=n,(TYP=(D2!D4!F2!F4),SOB=(adr!symb),DOB=(adr!symb)
!PARA=(adr!symb))[;Kommentar]

*** PAUS

1.1. Pausieren einer Task

```
1.2. RST   n
     DB    3   ;   Rufnummer
     DB    l   ;   Parameterlaenge
                   ( l=0   - Einfacher Ruf
                     l=80H - Ruf wird protokolliert )

1.3. PAUS   R=n[,PRT!PARA=(adr!symb)]
```

2.1. Pausieren einer Task ueber eine bestimmte Zeitdauer

```
2.2. RST   n
     DB    3   ;   Rufnummer
     DB    l   ;   Parameterlaenge
                   ( l=3   - Einfacher Ruf
                     l=83H - Ruf wird protokolliert )
     DB    ze  ;   Zeitbereich (HR=1,MI=2,SE=3,CL=4)
     DA    t   ;   Pausezeit

2.3. PAUS   R=n,(PTIM=t(HR!MI!SE!CL)[,PRT]
                !PARA=(adr!symb))[;Kommentar]
```

*** READ

1.1. Tastatureingabe

```
1.2. RST   n
     DB    8   ;   Rufnummer
     DB    7   ;   Parameterlaenge
     DB    m   ;   Parametermaske
     DB    i   ;   Geraetenummer
     DB    1   ;   Maschinenkommando  (k=1)
     DB    s   ;   Spaltenadresse
     DB    z   ;   Zeilenadresse
     DB    l   ;   Eingabelaenge
     DB    i   ;   Log. Geraetenummer des Ziel- MON1
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen.

Bit 0 : Geraetenr. MON1, Laenge, Spalte und Zeile
Bit 4 : Ruf mit WAIT

1.3. READ R=n,(KEYB=i,IN=k,BPOS=z/s,LOB=l,MON=i[,WAIT] !PARA=(adr!symb))[;Kommentar]

2.1. Tastatureingabe mit Transfer nach Pufferadresse

```
2.2. RST  n
     DB   8    ;   Rufnummer
     DB   9    ;   Parameterlaenge
     DB   m    ;   Parametermaske
     DB   i    ;   Log. Geraetenummer der Tastatur
     DB   2    ;   Maschinenkommando  (k=2)
     DB   s    ;   Spaltenadresse
     DB   z    ;   Zeilenadresse
     DB   l    ;   Eingabelaenge
     DB   i    ;   Log. Geraetenummer des Ziel- MON1
     DA  adr   ;   Pufferadresse der Task
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen.

Bit 0 : Geraetenr. MON1, Laenge, Spalte und Zeile
Bit 1 : Pufferadresse
Bit 4 : Ruf mit WAIT

2.3. READ R=n,(KEYB=i,IN=k,BPOS=z/s,LOB=l,MON=i, BOB=(adr!symb)[,WAIT] !PARA=(adr!symb))[;Kommentar]

3.1. Tastatureingabe eines Funktionswertes nach Pufferadresse

3.2.
```
RST   n
DB    8   ;   Rufnummer
DB    5   ;   Parameterlaenge
DB    m   ;   Parametermaske
DB    i   ;   Log. Geraetenummer der Tastatur
DB    4   ;   Maschinenkommando (k=3)
DA   adr  ;   Pufferadresse der Task
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen.

Bit 1 : Pufferadresse
Bit 4 : Ruf mit WAIT

3.3.
```
READ   R=n,(KEYB=i,IN=k,BOB=(adr!symb)
             !PARA=(adr!symb))[;Kommentar]
```

4.1. Datenuebertragung vom Folienspeicher zum Internspeicher

4.2.
```
RST   n
DB    8   ;   Rufnummer
DB    l   ;   Parameterlaenge (l=9 -Ruf ohne Fehlerschlues-
                                           sel
                                      l=11-Ruf mit  Fehlerschlues-
                                           sel)
DB    m   ;   Parametermaske
DB    i   ;   Log.Geraetenummer
DB    2   ;   Maschinenkommando fuer READ
DA   adr  ;   Zieladresse, Anfang logischer Puffer
DA    d   ;   Zu lesende Sektoranzahl
DA   tr   ;   Spur/Sektor
              ( Spur: 0 ≤ t ≤ 73 / Sektor: 1 ≤ r ≤ 26)
DA   adr  ;   Fehlerschluesseladresse (ungleich 0)
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

Bit 0 : Zieladresse und Sektoranzahl
Bit 1 : Erster zu lesender Sektor
Bit 2 : Fehlerschluesseladresse
Bit 4 : Ruf mit WAIT

4.3. READ R=n,(FD=i,BOB=(adr!symb),AOR=1,BPOS=t/r,
ECA=(adr!symb)[,WAIT]!PARA=(adr!symb))[;Kommentar]

5.1. Datenuebernahme vom Lochbandleser auf Internspeicher

```
5.2. RST  n
     DB   8   ;  Rufnummer
     DB   l   ;  Parameterlaenge
                 (l=9  - Zeichenuebernahme mit Angabe der Feh-
                         lerschluesseladresse
                  l=10 - Zeichenuebernahme bis Endezeichen mit
                         Angabe der Fehlerschluesseladresse )
     DB   m   ;  Parametermaske
     DB   i   ;  Logische Geraetenummer
     DB   mk  ;  Maschinenkommando
                 mk=01H:  Lesen ohne NUL und DEL  (k=1)
                 mk=03H:  Lesen mit NUL und DEL  (k=2)
                 mk=05H:  Rueckwaertslesen ohne NUL und DEL
                          (k=3)
                 mk=07H:  Rueckwaertslesen mit NUL und DEL
                          (k=4)
                 mk=11H:  Lesen ohne NUL und DEL mit Pari-
                          taetskontrolle  (k=5)
                 mk=13H:  Lesen mit NUL und DEL mit Paritaets-
                          kontrolle  (k=6)
                 mk=15H:  Rueckwaertslesen ohne NUL und DEL
                          mit Paritaetskontrolle  (k=7)
```

```
                 mk=17H:  Rueckwaertslesen mit NUL und DEL mit
                          Paritaetskontrolle  (k=8)
     DA   adr   ;   Pufferanfangsadresse
     DA     l   ;   Pufferlaenge
     DB     c   ;   Endezeichen
     DA    adr  ;   Fehlerschluesseladresse
     Parametermaske: Im Ruf enthaltene Parameter sind in der
                     Maske mit log. 1 zu kennzeichnen

                     Bit 0 : Pufferanfangsadresse und Puffer -
                             laenge
                     Bit 2 : Fehlerschluesseladresse
                     Bit 3 : Mit Angabe eines Endezeichens
                     Bit 4 : Ruf mit WAIT
```

```
5.3. READ    R=n,(PTR=i,IN=k,LOB=l,
                  BOB=(adr!symb)[,EOD=c][,ECA=(adr!symb)][,WAIT]
                 !PARA=(adr!symb))[;Kommentar]
```

*** RUN

1.1. Starten einer Task

```
1.2. RST   n
     DB    0    ;   Rufnummer
     DB    1    ;   Parameterlaenge
     DB    p    ;   Tasknummer
```

```
1.3. RUN   R=n,(TASK=p!PARA=(adr!symb))[;Kommentar]
```

2.1. Verzoegerter Start einer Task

```
2.2. RST   n
     DB    0    ;   Rufnummer
     DB    4    ;   Parameterlaenge
```

```
      DB    p   ;   Tasknummer
      DB    ze  ;   Zeitbereich (HR=1,MI=2,SE=3,CL=4)
      DA    t   ;   Verzoegerungszeit
```

```
2.3.  RUN   R=n,(TASK=p,DET=t(HR!MI!SE!CL)
                !PARA=(adr!symb))[;Kommentar]
```

3.1. Start einer Task zu einer gewuenschten Zeit

```
3.2.  RST   n
      DB    0   ;   Rufnummer
      DB    4   ;   Parameterlaenge
      DB    p   ;   Tasknummer
      DB    82H ;   Zeitbereich
      DB    hr  ;   Angabe der Stunden
      DB    mi  ;   Angabe der Minuten
```

```
3.3.  RUN   R=n,(TASK=p,RT=hr.mi!PARA=(adr!symb))[;Kommentar]
```

4.1. Zyklischer Start einer Task

```
4.2.  RST   n
      DB    0   ;   Rufnummer
      DB    84H ;   Parameterlaenge
      DB    p   ;   Tasknummer
      DB    ze  ;   Zeitbereich (HR=1,MI=2,SE=3,CL=4)
      DA    t   ;   Zykluszeit
```

```
4.3.  RUN   R=n,(TASK=p,CT=t(HR!MI!SE!CL)
                !PARA=(adr!symb))[;Kommentar]
```

5.1. Verzoegerter zyklischer Start einer Task

```
5.2.  RST   n
      DB    0   ;   Rufnummer
      DB    6   ;   Parameterlaenge
```

```
DB    p    ;    Tasknummer     ..................
DB    ze   ;    Zeitbereich    :  c t  :  d e t  :
                               :.......:.........:
                               B7    B4 B3      B0
                ( ct  - Zeitbereich fuer Zykluszeit
                  det - Zeitbereich fuer Verzoegerungszeit )
DA    t    ;    Verzoegerungszeit
DA    t    ;    Zykluszeit
```

```
5.3. RUN    R=n,(TASK=p,CT=t(HR!MI!SE!CL),DET=t(HR!MI!SE!CL)
                !PARA=(adr!symb))[;Kommentar]
```

6.1. Zyklischer Start einer Task zu einer gewuenschten Zeit

```
6.2. RST    n
     DB     0    ;    Rufnummer
     DB     6    ;    Parameterlaenge
     DB     p    ;    Tasknummer     .........................
     DB     ze   ;    Zeitbereich    : 1:    ct    : 0: 0: 1: 0:
                                     :..:.........:..:..:..:..:
                                      B7 B6 B5 B4 B3 B2 B1 B0
                      ( ct - Zeitbereich fuer Zykluszeit )
     DB     hr   ;    Angabe der Stunden
     DB     mi   ;    Angabe der Minuten
     DA     t    ;    Zykluszeit
```

```
6.3. RUN    R=n,(TASK=p,RT=hr.mi,CT=t(HR!MI!SE!CL)
                !PARA=(adr!symb))[;Kommentar]
```

*** TIME

1. Uebernahme der Zeit

```
2.   RST    n
     DB     12   ;    Rufnummer
```

```
     DB    2   ;   Parameteradresse
     DA   adr  ;   Zieladresse
```

3. `TIME    R=n,(BOB=(adr!symb)!PARA=(adr!symb))[;Kommentar]`

*** WAIT

1. Testen des Beenden einer E/A- Operation

2.
```
     RST   n
     DB   10   ;   Rufnummer
     DB    2   ;   Parameterlaenge
     DA   adr  ;   Adresse des E/A- Rufes
```

3. `WAIT    R=n,(CADR=(adr!symb)!PARA=(adr!symb))[;Kommentar]`

*** WRIT

1.1. Ausgabe von Zeichen auf Seriendrucker

1.2.
```
     RST   n
     DB    8   ;   Rufnummer
     DB    1   ;   Parameterlaenge
     DB    m   ;   Parametermaske
     DB    i   ;   Logische Geraetenummer
     DB   mk   ;   Maschinenkommando
                   mk=01H:  Zeichendruck   (k=1)
                   mk=81H:  Kombinierter Ruf   (k=2)
```

```
DA  adr   ;   Pufferanfangsadresse
DA    l   ;   Laenge des Druckpuffers  ( 0 ≤  l ≤  170 )
DA   po   ;   Positioniergroesse
              L-Teil von po  (bei Steuerfunktion)

               B7 B6 B5 B4 B3 B2 B1 B0
              .........................        Leporello  1
              : 0: 0: 0:     z e i     :    Zeilenschaltung
              :..:..:..:...............:           max  1FH

              .........................        Leporello  1
              : 0: 0: 1: 0: 0: 0: 0: 0:     Seitenschaltung
              :..:..:..:..:..:..:..:..:           einmalig

              .........................
              : 0: 1: 0:     z e i     :  Leporello 1 und 2
              :..:..:..:...............:    Zeilenschaltung

              .........................
              : 0: 1: 1: 0: 0: 0: 0: 0:   Leporello 1 und 2
              :..:..:..:..:..:..:..:..:    Seitenschaltung

              .........................
              : 1: 0: 1: 0: 0: 0: 0: 0:         Leporello 2
              :..:..:..:..:..:..:..:..:    Seitenschaltung

              .........................
              : 1: 0: 0:     z e i     :        Leporello 2
              :..:..:..:...............:    Zeilenschaltung

DA   adr  ;   Fehlerschluesseladresse

zei -  Anzahl der Zeilenschaltungen (maximal 31 Vorschuebe)
H-Teil von po  (bei Steuerfunktion) : 0
```

(bei Zeichendruck) : Position des ersten Druckzeichens
(bei Kombination) : wie H-Teil bei Zeichendruck

L-Teil von po (bei Zeichendruck) : 0
(bei Kombination) : wie L-Teil bei Steuerfunktion

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

Bit 0 : Pufferanfangsadresse und Puffer - laenge
Bit 1 : Positioniergroesse
Bit 2 : Fehlerschluesseladresse
Bit 4 : Ruf mit WAIT

1.3. WRIT R=n,(LP=i,IN=k,BOB=(adr!symb),LOB=l
[,BPOS=d][,FLF=s!z),LEP=(1!2!3)]
[,ECA=(adr!symb)]
[,WAIT]!PARA=(adr!symb))[;Kommentar]

2.1. Datenuebertragung vom Internspeicher auf Folienspeicher

2.2. RST n

```
DB   8    ;   Rufnummer
DB   l    ;   Parameterlaenge
              ( l=9  - Ruf ohne Fehlerschluessel
                l=11 - Ruf mit Fehlerschluessel  )
DB   m    ;   Parametermaske
DB   i    ;   Logische Geraetenummer
DB   1    ;   Maschinenkommando fuer WRIT
DA  adr   ;   Quelladresse, Anfang logischer Puffer
DA   a    ;   Zu schreibende Sektoranzahl
DB   r    ;   Sektor (erster Zielsektor)
DB   t    ;   Spur (erste Zielspur)
DA  adr   ;   Fehlerschluesseladresse
```

```
Parametermaske: Im Ruf enthaltene Parameter sind in der
                Maske mit log. 1 zu kennzeichnen

                Bit 0 : Zieladresse,Sektoranzahl
                Bit 1 : Erster zu schreibender Sektor
                Bit 2 : Fehlerschluesseladresse
                Bit 4 : Ruf mit WAIT
```

```
2.3. WRIT   R=n,(FD=i,BOB=(adr!symb),AOR=l,BOD=t/r,
                  [ECA=(adr!symb)][,WAIT]
                 !PARA=(adr!symb))[;Kommentar]
```

3.1. Ausgabe auf MON1 beginnend ab Nutzeranfang

```
3.2. RST    n
     DB     8   ;   Rufnummer
     DB     l   ;   Parameterlaenge  (l=7 - ohne Fehlerschluessel
                                      l=9 -  mit Fehlerschluessel)
     DB     m   ;   Parametermaske
     DB     i   ;   Logische Geraetenummer MON1
     DB    mk   ;   Maschinenkommando
                    mk=2:   Zeichenausgabe mit Angabe der letzten
                            Zielposition in Zeile und Spalte
                            (k=1)
                    mk=10:  Zeichenausgabe mit Angabe der  Anzahl
                            der auszugebenden Zeichen  (k=2)
     DA   adr   ;   Quelladresse, logischer Puffer
     DA (zs!l)  ;   Endposition oder Ausgabelaenge
                    mk=02H: Endposition in Zeile und Spalte
                    mk=0AH: Ausgabelaenge
     DA    adr  ;   Fehlerschluesseladresse
```

```
Parametermaske: Im Ruf enthaltene Parameter sind in der
                Maske mit log. 1 zu kennzeichnen
```

```
                         Bit 0 : Quellfeldadresse (Endposition  oder
                                 Laenge)
                         Bit 2 : Fehlerschluesseladresse
                         Bit 4 : Ruf mit WAIT

3.3. WRIT    R=n,(MON=i,IN=k,(EPOS=z/s!LOB=l),
                  BOB=(adr!symb),ECA=(adr!symb)[,WAIT]
                 !PARA=(adr!symb))[;Kommentar]

4.1. Ausgabe auf MON1 ab Beginnposition

4.2. RST    n
     DB     8    ;   Rufnummer
     DB     l    ;   Parameterlaenge (l=9  - ohne Fehlerschluessel
                                      l=11 - mit Fehlerschluessel)
     DB     m    ;   Parametermaske
     DB     i    ;   Logische Geraetenummer MON1
     DB     k    ;   Maschinenkommando fuer positionierte Zeichen-
                     ausgabe
                     mk=06H: Zeichenausgabe mit Angabe der letzten
                             Zielposition in Zeile und Spalte(k=3)
                     mk=0EH: Zeichenausgabe mit der Anzahl  auszu-
                             gebender Zeichen  (k=4)
     DA   adr    ;   Quellfeldadresse, logischer Puffer
     DA (zs!l)   ;   Endposition oder Ausgabelaenge
                     mk=06H: Zeile und Spalte der Beginnposition
                     mk=0EH: Ausgabelaenge
     DA    zs    ;   Anfangsposition
     DA    adr   ;   Fehlerschluesseladresse

     Parametermaske: Im Ruf enthaltene  Parameter  sind  in  der
                     Maske mit log. 1 zu kennzeichnen
                         Bit 0 : Quellfeldadresse (Endpos. , Laenge)
                         Bit 1 : Anfangsposition auf MON1
                         Bit 2 : Fehlerschluesseladresse
                         Bit 4 : Ruf mit WAIT

4.3. WRIT    R=n,(MON=i,IN=k,BOB=(adr!symb),BPOS=z/s,
                 (EPOS=z/s!LOB=l),ECA=(adr!symb)
                 !PARA=(adr!symb))[;Kommentar]
```

5.1. Ausgabe von Zeichen auf den Lochbandstanzer

5.2.
```
RST   n
DB    8    ;   Rufnummer
DB    1    ;   Parameterlaenge
               (1=9  - Zeichenuebertragung mit Angabe der
                       Fehlerschluesseladresse
                1=10 - Zeichenuebertragung bis Endezeichen
                       mit Angabe der Fehlerschluesseladres-
                       se)
DB    m    ;   Parametermaske
DB    i    ;   Logische Geraetenummer
DB    mk   ;   Maschinenkommando
               mk=01H:  Stanzen mit automatischer Aufzeich-
                        nungswiederholung (k=1)
               mk=05H:  Stanzen ohne automatische Aufzeich-
                        nungswiederholung (k=2)
DA    adr  ;   Pufferanfangsadresse
DA    1    ;   Pufferlaenge
DA    c    ;   Endezeichen
DA    adr  ;   Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

- Bit 0 : Pufferanfangsadresse und Puffer - laenge
- Bit 2 : Fehlerschluesseladresse
- Bit 3 : Mit Angabe eines Endezeichens
- Bit 4 : Ruf mit WAIT

5.3.
```
WRIT   R=n,(PTP=i,IN=k,BOB=(adr!symb),LOB=1[,EOD=c]
                 [,ECA=(adr!symb)][,WAIT]
                !PARA=(adr!symb))[;Kommentar]
```

***** Rufe des FILE-Handlers *****

In den FILE-Handler- Rufen gilt :

name (Name der Datei)
- 4 alphanumerische Zeichen
- erstes Zeichen muss ein Buchstabe sein

*** OPEN

1. Eroeffnung von Dateien

2.
```
RST  n
DB   11     ;  Rufnummer
DB   l      ;  Parameterlaenge
               (l=9  - ohne Anfangswert
                l=11 - mit Anfangswert )
DB   2      ;  Rufnummer OPEN
DB   m      ;  Parametermaske
DB   j      ;  Laufwerknummer
DB 'name'   ;  Dateiname
DA   z      ;  Anfangswert
DA  adr     ;  Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

Bit 5 : Anfangswert
Bit 6 : Schreibschutz
Bit 7 : Dateityp (Bit wird gesetzt, wenn TY = S)

3.
```
OPEN    R=n,(NAME='name',TYP=(S!D),FD=j[,FDS=z][,WRP],
             ECA=(adr!symb)!PARA=(adr!symb))[;Kommentar]
```

*** CLOS

1. Schliessen von Dateien

2.
```
RST   n
DB    11     ;   Rufnummer
DB     7     ;   Parameterlaenge
DB     2     ;   Rufnummer CLOSE
DB 'name'    ;   Dateiname
DA   adr     ;   Fehlerschluesseladresse
```

3.
```
CLOSE   R=n,(NAME='name',ECA=(adr!symb)
             !PARA=(adr!symb))[;Kommentar]
```

*** POSF

1. Positionieren der Datei auf Anfang

2.
```
RST   n
DB    11     ;   Rufnummer
DB     7     ;   Parameterlaenge
DB     3     ;   Rufnummer POSF
DB 'name'    ;   Dateiname
DA   adr     ;   Fehlerschluesseladresse
```

3.
```
POSF   R=n,(NAME='name',ECA=(adr!symb)
            !PARA=(adr!symb))[;Kommentar]
```

*** FIDE

1. Eroeffnung der Dateierklaerung

2.
```
RST   n
DB    11     ;   Rufnummer
DB    12     ;   Parameterlaenge
```

```
DB    4     ;    Rufnummer FIDE
DB    j     ;    Laufwerknummer
DB 'name'   ;    Dateiname
DA    l     ;    Satzlaenge
DA    a     ;    Satzanzahl
DA  adr     ;    Fehlerschluesseladresse
```

3. FIDE R=n,(NAME='name',FD=j,LDS=l,ADS=a,ECA=(adr!symb)
   !PARA=(adr!symb))[;Kommentar]

*** CFID

1. Schliessen der Dateierklaerung

2. RST n

```
DB   11     ;    Rufnummer
DB    8     ;    Parameterlaenge
DB    5     ;    Rufnummer CFID
DB    j     ;    Laufwerknummer
DB 'name'   ;    Dateiname
DA  adr     ;    Fehlerschluesseladresse
```

3. CFID R=n,(NAME='name',FD=j,ECA=(adr!symb)
   !PARA=(adr!symb))[;Kommentar]

*** DISK

1. Lesen/Schreiben in der Datei

```
2.   RST    n
     DB    11    ;   Rufnummer
     DB     l    ;   Parameterlaenge
                     (l=12 -ohne Angabe der Satznummer
                      l=14 -mit Angabe der Satznummer)
     DB     6    ;   Rufnummer DISK
     DB     m    ;   Parametermaske
     DB 'name'   ;   Dateiname
     DA     s    ;   Satznummer (entfaellt bei TYP=S)
     DA     a    ;   Satzanzahl
     DA   adr    ;   Pufferanfangsadresse
     DA   adr    ;   Fehlerschluesseladresse
```

Parametermaske: Im Ruf enthaltene Parameter sind in der Maske mit log. 1 zu kennzeichnen

Bit 4 : bei k=2
Bit 5 : bei TYP=S
Bit 6 : Satznummer symbolisch angegeben
Bit 7 : Satzanzahl symbolisch angegeben

```
3.   DISK    R=n,(NAME='name',IN=k,TYP=(S!D),
                  NDS=(s!symb),ADS=(a!symb),BOB=(adr!symb),
                  ECA=(adr!symb)!PARA=(adr!symb))[;Kommentar]
```

Anlage 2:   Uebersicht der EIEX-Kommandos

Jedes EIEX-Kommando wird durch folgende 3 Punkte erlaeutert:

*** Kommandoname
1. Bedeutung
2. Kommandonotation

*** GO

1. Fortsetzen einer pausierenden Task
2. GO TASK=p

*** DISP

1. Verhindern der Bearbeitung einer Task
2. DISP TASK=p

*** ENAP

1. Erlauben der Bearbeitung einer verhinderten Task
2. ENAP TASK=p

*** CNCL

1.1. Abbruch einer Task
1.2. CNCL TASK=p
2.1. Austragen einer Task aus der Zeitorganisation
2.2. CNCL TASK=p, TM
3.1. Austragen aller Task aus der Zeitorganisation
3.2. CNCL TASK=ALL, TM

*** RUN

1.1. Starten einer Task
1.2. RUN TASK=p
2.1. Verzoegerter Start einer Task
2.2. RUN TASK=p, DET=t(HR!MI!SE!CL)
3.1. Verzoegerter zyklischer Start einer Task
3.2. RUN TASK=p, DET=t(HR!MI!SE!CL), CT=t(HR!MI!SE!CL)

4.1. Start einer Task zu einer gewuenschten Zeit
4.2. RUN TASK=p, DET=, RT=hr.mi
5.1. Zyklischer Start einer Task
5.2. RUN TASK=p, DET=, RT=, CT=t(HR!MI!SE!CL)
6.1. Zyklischer Start einer Task zu einer gewuenschten Zeit
6.2. RUN TASK=p, DET=, RT=hr.mi, CT=t(HR!MI!SE!CL)

*** STIME
1. Stellen der Echtzeituhr
2. STIME=hr.mi

*** SDATE
1. Stellen des Kalenders
2. SDATE=da.mo.ye

*** LOG
1. Einschalten Protokollierung Drucker
2. LOG

*** NOLOG
1. Ausschalten Protokollierung Drucker
2. NOLOG

*** HELP
1. Wechsel der Prioritaet einer Task
2. HELP TASK=p

*** CHAN
1.1. Wechsel der Prioritaet zweier Tasks
1.2. CHAN TASK=p, CTASK=p [,BYE]
2.1. Ruecksetzen gewechselter Prioritaeten zweier Tasks
2.2. CHAN TASK=p

Anlage 3: Fehlerausschriften und **Systemnachrichten** EIEX 1521

SKZ	FNR	Fehlerausschrift/ Systemnachricht	Fehlerursache	Reaktion EIEX 1521	IHF	Reaktion des Anwenders
B	1	INVALID STATEMENT ON MON = xx	siehe K 1	siehe R 6	10H	siehe K 1
B	2	PROTECTED USER AREA ON DEVICE xx	Bereichsueberschreitung	E/A-Ruf zurueckgestellt und nach Tastatureingabe nachgestartet	80H	Tastatureingabe beenden
C	1	DEVICE ERROR ON DEVICE xx	Folgeinterrupt nicht innerhalb normaler Reaktionszeit	Taskfortsetzung der Ruf wird nach Einschalten des Geraetes abgearbeitet	22H	Geraet wechseln bzw einschalten
C,D	1	DEVICE ERROR ON DEVICE xx	Statusfehler (Geraetedefekt)	siehe R 6	20H	bei gener. Ersatzgeraet Ger.-umschaltung durch Ruf o. Kommando

D	2	DEVICE ADDRESS P OUT OF SYSTEM	Im Ruf stehende Geraetenummer liegt ausserhalb der gener. log. Geraete	siehe R 6	11H	siehe K 1
D	3	USER ERROR ON DEVICE xx	Statusfehler (Fehlbedienung)	siehe R 6	40H	Fehlerquelle beseitigen, Ausfuehrung der Geraete-rueckschaltung
D	4	DEVICE EXCHANGE NOT POSSIBLE	autom. Geraeteumschaltung nicht moeglich- kein verfuegbares aktuelles Ersatzgeraet	siehe R 6	20H	siehe D 2
D	5	DEVICE xx NOT AVAILABLE	zyklischer E/A-Ruf auf Defektgeraet bzw uneingeschaltetes Geraet	E/A-Ruf wird uebergangen, Task bleibt in WAIT	-	Geraet einschalten bzw austauschen

D	19	DEVICE xx WARNING DATA MIX	autom. Geraeteumschaltung auf ein Geraet, das bereits gearbeitet hat	AGU und Rufausfuehrung		
D	20	EXCHANGE OF DEVICE xx AND yy OK!		AGU ausgefuehrt	–	
E	1	OVERFLOW IN ERROR MANAGEMENT	Die Anzahl der aufzubereitenden Fehlermeldungen ueberschreitet die generierte Puffertiefe	Der Fehlerpuffer wird komplett geloescht	–	
F	1	SEEK CHECK ON FD = xx TRK = xx,SEC = yy	ID-Feld nicht lesbar (Datenverlust)	Taskfortsetzung	80H	evtl. Laufwerk wechseln
F	2	READ CHECK ON FD = xx TRK = xx,SEC = yy	CRC-Fehler/Datenmarkenausfall (Datenverlust)	Taskfortsetzung	81H	siehe F 1
F	3	SEEK CHECK ON FD = xx TRK = xx,SEC = yy	Kopfpositionierung nicht moeglich (kein Datentransfer) Geraet defekt!	Taskfortsetzung	20H	Laufwerk wechseln!

F	4	WRONG OPERATOR INTER-VENTION ON FD = xx 'R'	Drehzahlfehler (kein Datentrans-fer)	Taskfortsetzung	40H	Laufwerk wechseln!
			Bedienerfehler			nochmalige richtige Be-dienung
F	5	WARNING ON FD = xx - COPY = FD 'R'	Vorwarnung-Diskette duplizieren	Taskfortsetzung	41H	Diskette duplizieren
F	6	INTERVENTION REQUEST ON FD = xx - COPY = FD 'R'	Aufforderung-Dis-kette duplizieren	Taskfortsetzung	42H	siehe F 5
F	7	WRIT DATA CHECK-ERMAP OVERFLOW ON FD = xx	Errormapueberlauf (Datentransfer nicht moeglich)	Taskfortsetzung	82H	Diskette verwerfen
F	8	READ DATA CHECK-ERMAP ON FD = xx	Errormap nicht les-bar (Datenverlust)	Taskfortsetzung	83H	siehe F 7
F	9	WRIT DATA CHECK-ERMAP ON FD = xx	Aktualisierung Er-rormap fehlerhaft (Datenverlust)	Taskfortsetzung	82H	siehe F 7

F	10	WRIT DATA CHECK-ERSET ON FD = xx	(Sektorauslagerung nicht moeglich) Writ Errorset fehlerhaft	Taskfortsetzung	82H	siehe F 7
F	11	INVALID STATEMENT ON FD = xx TASK xx 'R'	siehe K 1	siehe R 6	10H	siehe K 1
H	1	FILE AREA OVERFLOW	Freier Nutzerbereich Diskette ist kleiner als benoetigter Nutzerbereich	Setzen Fehlerzelle Ruf und Ruf beenden	41H	E/A-Ruf korrigieren
H	2	FILE NOT OPENED	Angesprochene Datei ist nicht eroeffnet	Setzen Fehlerzelle Ruf und Ruf beenden	42H	Datei eroeffnen
H	3	ON DEVICE xx OUT OF SYSTEM	logische Laufwerknummer nicht vorhanden	Setzen Fehlerzelle Ruf und Ruf beenden	43H	siehe H 1
H	4	FILE PROTECTED	Schreiben Datei bei gesetztem Schreibschutz	Setzen Fehlerzelle Ruf und Ruf beenden	44H	siehe H 1

H	5	RECORD NUMBER ERROR	Angegebene Satznummer groesser als maximale Satzanzahl Datei	Setzen Fehlerzelle Ruf und Ruf beenden	45H	siehe H 1
H	6	VOLUME LABEL ERROR ON DEVICE xx	Diskette im logischen Laufwerk DEVICE = x besitzt keinen Datentraegerkennsatz	setzen Fehlerzelle Ruf und Ruf beenden	46H	Datentraegerkennsatz vereinbaren
H	7	FILE NAME ALREADY IN DIRECTORY	Dateiname schon vorhanden	setzen Fehlerzelle Ruf und Ruf beenden	47H	siehe H 1
H	8	FILE NOT FOUND	Dateiname unbekannt	setzen Fehlerzelle Ruf und Ruf beenden	48H	siehe H 1 Diskette wechseln
H	9	FILE CONTROL BLOCK OVERFLOW	Generierte maximale Anzahl interner Dateisteuerbloecke ueberschritten	setzen Fehlerzelle Ruf und Ruf beenden	49H	Fehler bei der Generierung

H	10	ON DEVICE xx INDEX TRACK OVERFLOW	Dateikennsatz log. Laufwerk DEVICE = x belegt	setzen Fehlerzelle Ruf und Ruf beenden	4AH	Diskette wechseln
I	1	CALL n OUT OF SYSTEM	Ruf-Nr. unzulaessig Ruf-Routine nicht generiert	siehe R 1	–	siehe R 1
K	1	INVALID STATEMENT ON DEVICE xx	Fehlerhafter Rufaufbau	siehe R 6	10H	Ruf im AP-System korregieren
(K)	–	COMMAND ERROR 1	Kommando-Wort in EIEX nicht definiert	–	–	DEL-Taste, Kommandowort neu eingeben
(K)	–	COMMAND ERROR 2	Kein numerischer Wert	–	–	DEL-Taste, neue Eingabe
(K)	–	COMMAND ERROR 3	Wert zu gross/zu klein	–	–	DEL-Taste, Zeit-Eingabe wiederholen
(K)	–	COMMAND ERROR 4	Zeit-Einheit ungleich HR,MI,SE oder CL	–	–	DEL-Taste, Zeit-Eingabe

(K)	-	COMMAND ERROR 5	Trennzeichen falsch	-	-	DEL-Taste, Eingabe wiederholen
(K)	-	COMMAND ERROR 6	Unzulaessige Task-Nr.	-	-	DEL-Taste, Task-Nr. neu eingeben
(K)	-	COMMAND ERROR 7	Fehlerhafte Bedienungsfolge	-	-	DEL-Taste, Eingabe wiederholen
L	1	INVALID STATEMENT ON DEVICE xx	siehe K 1	siehe R 6	10H	siehe K 1
L	2	PARITY ERROR xx	Fehlerhafte Datenträgerlochung Ruf wird abgearbeitet	Reaktion im Treiber (Absetzen eines Fehlerzeichens H T	80H	
P	1	DEVICE xx OUT OF SYSTEM	logische Geraete-Nr. nicht zulaessig	siehe R 6	42H	siehe V 1

P	2	DEVICE xx UNKNOW	zum logischen Geraet ist keine Zuweisungstabelle gener.	siehe R 6	43H	siehe V 1
P	3	OCCUPIED DEVICE xx TASK STOPED	logisches Geraet besetzt, Task niederer Prioritaet wird gestoppt	Task wird zeitlich unterbrochen und durch EO erneut gestartet	–	
P	4	OCCUPIED DEVICE xx TASK WAITING	logisches Geraet besetzt, Task hoeherer Prioritaet traegt seine Anforderung im Wartepuffer ein und geht in Wartezustand	Task wird zeitlich unterbrochen und durch EV verwaltet -zusaetzliche Systembelastung und evtl. Systemblokkierung	–	
P	5	INVALID STATEMENT ADRESS UNKNOW	Port-Nr. Angabe in Tabelle der neuen Impulszeitkonstanten unzulaessig		45H	
P	6	ADRESS UNKNOW	Umwandlung der Port-Nr. in zugehoerige Impulsgeber-Adresse nicht moeglich		46H	

P	7	NOT POSSIBLE COMMAND	Fehler in den Initialisierungsparametern fuer die Prozess-Steckkarte DEAS:	
		01	unzulaessiges Status-Wort fuer Byte-Betrieb	
		02	unzulaessiges 1.Byte in Port-Initialisierungsparameter (nur 03, Vektor oder Interruptsteuerwort zulaessig)	
		03	Bei freigegebenem Interrupt keine Interruptmaske angegeben	
		04	Parameter nach Interruptmaske kein Vektor	
		05	Bei bidireaktionalem Betrieb fehlender oder unzulaessiger Vektor	
P	8	OVERFLOW IN TIME MANAGEMENT	Generierte Anzahl der Geraete der Prozessperipherie zu klein	
P	9	xx ERROR	Anzahl der Initialisierungsfehler der Prozessperipherie	
P	10	DEVICE xx ADRESS ERROR	Fehlerhafte Kanal-Adresse	10H
P	11	ERROR ON DEVICE xx	Fuer die Typen DAS werden die fehlerhaften Kanaele mit "1" in der zugeordneten Bitposition gekennzeichnet	=0

			Fehlerhafte Arbeit des log.Geraetes bei Geraetekontrolle erkannt		20H	
P	12	DEVICE xx STOPED	Kartenstatus "Ausser Funktion"		24H	
R	1	TASK p OUT OF SYSTEM	Task p liegt ausserhalb der generierten Task	Ruf wird nicht ausgefuehrt, Task bzw. ISR wird fortgesetzt	–	generiertes SPS pruefen, Ruf im AP-System bzw. in ISR korrigieren
R	2	TASK p PROTECTED	Task p ist als Systemtask definiert	siehe R 1	–	siehe R 1
R,T	3	TASK p DISABLE	Task p ist fuer Abarbeitung gesperrt	siehe R 1	–	AP-System pruefen
R	4	Task p DOES NOT PAUSE	Task p befindet sich nicht im Pausezustand	siehe R 1	–	siehe R 3
R	5	TASK p ALREADY IN TIME MANAGEMENT	Task p wird bereits in der Zeitorganisation verwaltet	siehe R 1	–	siehe R 3

R	6	INVALID STATEMENT	Im Ruf stehende Adresse bezieht sich nicht auf einen E/A-Ruf	Ruf wird nicht ausgefuehrt, Task wird fortgesetzt	–	siehe R 3
			bzw. siehe K 1		10H	
			Konvertierungsmodul nicht vorhanden			
R	7	TASK LIMIT OVERFLOW IN TIME MANAGEMENT	Anzahl der generierten Task der Zeitorganisation wird ueberschritten	Ruf wirkungslos – rufende Task wird abgebrochen	–	siehe R 3
R	8	TASK p PAUSE	Protokollausschrift fuer pausierende Task		–	
R	9	INVALID CALL CHAN	Fehlerhaftes Ruflaengenbyte im Ruf CHAN	Der Ruf wird uebergangen und die Task fortgesetzt	–	siehe R 3

R	10	TASK p CHANGED	Die Prioritaet einer mit HELP oder CHAN verschobenen Task taucht als Parameter im Kommando HELP, bzw. Ruf oder Kommando CHAN auf.	siehe R 10	–	siehe R 3
R	11	OCCUPIED COMMAND HELP	Es wurde versucht, waehrend das Kommando HELP noch fuer eine Task wirksam ist, erneut das Kommando zu geben.	Das Kommando wird ignoriert	–	
R	12	TASK p NOT CHANGED	Der Ruf oder das Kommando CHAN TASK=p wird fuer eine Task gegeben, fuer die kein staendiger Prioritaetswechsl vorliegt.	Der Ruf wird uebergangen und die Task fortgesetzt	–	siehe R 3
R	13	BUFFER OVERFLOW CHAN	Der Ruf oder das Kommando CHAN werden gegeben, obwohl der Puffer voll ist.	siehe R 13	–	siehe R 3

R	14	TASK p PROTECTED	Im Ruf oder Kommando CHAN taucht die fuer das Kommando HELP reservierte Prioritaet auf	siehe R 13	-	siehe R 3
R	15	WRONG CODE-TABLE	Code-Tabelle steht auf unzulaessiger Adresse	siehe R 6	-	siehe R 3
R	16	DV-SET NOT POSSIBLE	Austauschgeraet ist aktuell besetzt	Ruf oder Kommando wird uebergangen	20H	Ruf oder Kommando neu starten
R	17	DEVICE xx yy CHANGED		Geraetezuweisung ausgefuehrt	-	
R	18	DEVICE EXCHANGE NOT POSSIBLE	Ersatzgeraet ist nicht verfuegbar	siehe R 6	20H	siehe R 16
R	19	DEVICE xx WARNING DATA MIX	Ersatzgeraet hat bereits gearbeitet	siehe R 6	11H	Ruf kann erneut gestartet werden - event. Datenmix
R	20	EXCHANGE OF DEVICE xx AND yy OK!		Geraeteumschaltung ausgefuehrt	-	

SKZ	FNR	Meldung	Bedeutung	Reaktion	IHF	Maßnahme
R	21	DV-RESET DEVICE xx OK!	Geraeterueckschaltung erfolgreich ausgefuehrt			
S	1	INVALID STATEMENT ON DEVICE xx	siehe K 1	siehe R 6	10H	siehe K 1
S	2	WARNING ON DEVICE xx END OF TAPE	Vorwarnung-Ende des Datentraegers	Bis nachfolgende Ausgabe von 1024 Zeichen wird Ruf abgearbeitet, sonst Abbruch	80H	neuen Datentraeger bereitstellen u. zur gegebenen Zeit auswechseln
V	1	TASK p OUT OF SYSTEM	Task wurde nicht generiert, Startadresse unbekannt	Task p wird nicht bearbeitet	– –	Generiertes SPS pruefen, AP-System korrigieren

SKZ – Systemkennzeichen
FNR – Fehlernummer
IHF – Inhalt der Fehlerzelle

Anlage 4: Schluesselwortparameter fuer Rufe und Kommandos

Schluessel-wort	Bedeutung	Geltungsbereich des zugeh. Parameters
AA1K	Analoge Ausgabe 1 Kanal	
AA5K	Analoge Ausgabe 5 Kanaele	
ADS	Satzanzahl der Datei (AMOUNT OF DATA SET)	$0 < a < 65536$
AEG	Analoge Eingabe	
ALL	Alle zeitlichen Task austragen	
AOR	Sektoranzahl (AMOUNT OF RECORD)	$0 < d < 256$
BOB	Pufferanfangsadresse (BEGIN OF BUFFER)	$0 <$ adr $<$ FFFFH symb
BOT	Adr. der Liste Impulszeitkonstanten	$0 <$ adr! $<$ FFFFH symb
BPOS	Beginnposition (BEGIN POSITION) a) Drucker b) Bildschirm c) Floppy	 $0 < d < 171$ $0 < z < 17$ $0 < s < 65$ $0 \leq t \leq 73$ $0 < r < 27$
BREAK	Zugriff auf besetztes Geraet	

BYE	Taskende	
CADR	Anfangsadr.des zugehoerigen E/A-Rufes	0 < adr < FFFFH symb
COTA	Anfangsadr. der CODE-Tabelle	modulo 256 0 < adr! < FFFFH symb
CT	Zykluszeit (CYCLE TIME)	0 < t < 32768
CTASK	Wechseltask (CHANGE TASK)	0 < p < 256
DAC	Digitale Ausgabe statisch (mit Eingabe)	
DAD	Digitale Ausgabe Dynamisch	
DAS	Digitale Ausgabe Statisch	
DEC	Digitale Eingabe Statisch (mit Ausgabe)	
DED	Digitale Eingabe Dynamisch	
DES	Digitale Eingabe Statisch	
DESM	Digitale Eingabe Statisch ueber Multiplexer	
DESU	Digitale Eingabe Statisch mit Unterbrechung	
DET	Verzoegerungszeit (DELAY TIME)	0 < t < 32768

DOB	Pufferadresse Zielbereich (DESTINATION BUFFER)	0 < adr! < FFFFH symb
ECA	Fehlerschluesseladresse (ERROR CONDITION ADDRESS)	0 < adr! < FFFFH symb
EOD	Endezeichen (END OF DATA)	c = Beliebiges ISO-Zeichen
EPOS	Endposition (END POSITION)	s / z wie bei BPOS
FD	Log.Geraetenummer Folienspeicher (FLOPPY DISK)	0 < i < 256
FDS	Anfangswert Satzzaehler Datei (FIRST DATA SET)	0 < z < 32768
FLF	Formulartraegersteuerung (FORM LINE FEED)	s = 1 0 < z < 32
IA	Impulsausgabe	
IN	Kommando (INSTRUCTION)	siehe Anlage 1 (Uebersicht der EIEX-Rufe)
KEYB	Logische Geraetenummer Tastatur (KEYBOARD)	0 < i < 256
LDS	Satzlaenge der Datei (LENGH OF DATA SET)	0 < l < 3328
LEP	Wahl des Leperello	1,2,3

LIB	Symbolischer Unterprogrammname (LIBRARY)	SR.EP Programmname Eintrittspunkt
LOB	Pufferlaenge (LENGTH OF BUFFER) a) Bildschirm b) Drucker c) LBL und LBS d) Tastatur e) Floppy	0 < l < FFFFH 0 < l < 400H 0 < l < 171 0 < l < FFFFH 0 < l < 256 0 < l < 257
LP	Logische Geraetenummer Drucker (LINE PRINTER)	0 < i < 256
MASK	Ausgabe-Maskenbyte	0 ≤ m ≤ 255
MASTER	Port-Nr. des steuernden Masters	0 < s < 256
MEMORY	Speicher fuer einlaufende Impulse	0 < adr! < FFFFH symb
MON	Logische Geraetenummer Monitor (MONITOR)	0 < i < 256
MUX	Adr. der Geraeteliste fuer MUX-Adressen	0 < adr! < FFFFH symb
NAME	Name der Datei (NAME)	Max. 4 Alphanum. Zeichen (1.Zeichen Buchstabe)
NDS	Satznummer (NUMBER OF DATA SET)	0 < s! < 65536 symb.

PARA	Indirekte Parameteradresse (PARAMETER ADDRESS)	0 < adr! < FFFFH symb
POM	Zeiger auf Maskenbereich	0 < adr! < FFFFH symb
POT	Adresse der Liste Takt-zeiten	0 < adr! < FFFFH symb
PRT	Protokollierungsanweisung (PRINT)	
PTIM	Pausezeit (PAUS TIME)	0 < t < 32768
PTP	Logische Geraetenummer Lochbandstanzer (PAPERTAPE PUNCH)	0 < i < 256
PTR	Logische Geraetenummer Lochbandleser (PAPERTAPE READ)	0 < i < 256
R	Registerrettungsart	0 ≤ n ≤ 2
REC	Anzahl der Datensaetze	0 < r < 256
REP	Mit Wiederstart (REPEAT)	
RT	Startzeit (REAL TIME)	hr.mi 0 ≤ hr ≤ 23
SCV	Geraete-Kontrollwort	

SEQ	Ersatzgeraet (STANDBY EQUIPMENT)	0 < i < 256
SERO	Adresse der Statusfehler- Routine	0 < adr! < FFFFH symb
SIZE	Blocklaenge	0 < m < 256
SOB	Pufferadresse Quellbereich (SOURCE OF BUFFER)	0 < adr < FFFFH
SUD	Defektgeraet (SUSPENDED DEVICE)	0 < i < 256
TAKT	Wert der Zeitkonstante	0 < t < 256
TASK	Tasknummer (TASK NUMBER)	0 < p < 256
TM	Austragen einer zeitlichen Task (TIME MANAGEMENT)	
TYP	a) Konvertierung Typ der Konvertierung b) Filehaendler	F1 = D2,D4,F2,F4 D,S
UIZ	Universeller Impulszaehler/ Zeitgeber	
USER	Adresse der Anwender-ISR	0 < adr! < FFFFH symb
WAIT	WAIT-Funktion (WAIT FUNCTION)	
WRP	Schreibschutz (WRITE PROTECTED)	

Anlage 5: Prozess-Steckkarten ursadat 5000

Typenbezeichnung	Funktion	Ausgang/Eingang
1. Digitale Ein-/ Ausgabe		
1.1. DAS-H 2330	Ausgabe statischer Signale	Haftrelais, Wechsler, 8 Bit
1.2. DAS-KT 2334	Ausgabe statischer Signale	Kurzschlussfeste Treiber KTSE-Ausgang, 4 x 8 Bit
1.3. DA-R 2331	Programmierbare Ausgabe statischer/dynamischer Signale in Stufen zu 8 Bit	Relais, 3 x 8 Bit
1.. D-T 2336	Programmierbare Ausgabe statischer/dynamischer Signale in Stufen zu 8 Bit	Schalttransistor, 4 x 8 Bit
1.5. DA-0 2335	Programmierbare Ausgabe statischer/dynamischer Signale in Stufen zu 8 Bit	Optokoppler, 2 x 8 Bit
1.6. DEAS 2337	Programmierbare Aus-/Eingabe von binaeren Signalen in Stufen zu 4 Bit, Handshakeleitung der PIO nach aussen zum Aufbau von Anschlusssteuerungen nach Quittungsprinzip	TTL-Pegel, 3 x 8 (2 x 4) Bit
1.7. DES 2340	Eingabe statischer Signale (Polling-oder Interruptbetrieb)	TTL-Pegel 5/12/24 (60)V, 2 x 8 Bit
1.8. DES-KT 2344	Eingabe statischer Signale (Polling)	KTSE-Eingang, 4 x 8 Bit

1.9. DED 2342	Eingabe dynamischer Signale (Interrupt)	TTL-Pegel 5/12/24 (60)V, 2 x 8 Bit
1.10. DEM 2341	Multiplexer (zyklische) Abfrage passiver Geber (Polling)	16 Ausgaenge 8 Eingaenge 1 Steuereingang 12/24 (60)V,max. 128 Geber
1.11. UIZ 2343	Impulszaehler, Frequenzmesser, Zeitimpulsgeber, Zeitmesser	TTL-Pegel 5/12/24 (60)V, 4 Torungseingaenge 4 Zaehleingaenge, 1 Zeitimpulsgeberausgang
1.12. IA 2339	Programmierbar als Impulsausgabe oder Zeitsignalausgabe	TTL-Pegel 5/12 V 2 unabhaengige Systeme
2. Analoge Ein-/ Ausgabe		
2.1. AA-1K 2301	Ausgabe analoger Signale, 1 Kanal (10/12 Bit)	unipolar/bipolar, U/I-Signale
2.2. AA-5K 2303	Ausgabe analoger Signale, 5 Kanaele (8 Bit)	unipolar/bipolar, U/I-Signale
2.3. AE-G 2310	Eingabe analoger Signale, (12 Bit), Grundkarte	bipolar, 1V, 8 Kanaele
2.4. AE-E 1) 2311	Expanderkarte	bipolar, 1V, 24 Kanaele
2.5. AE-TV 1) 2313	Trennverstaerker	4 Kanaele
2.6. AE-EV 1) 2314	Einzelverstaerker	4 Kanaele, < 1V
2.7. AE-PG 1) 2316	Anpassungskarte, passive Geber	4 Kanaele, Widerstandsthermometer Widerstandsferngeber

2.8. AE-AG 2315	Anpassungskarte, aktive Geber	8 Kanaele, 1/10V 5/10/20 mA
3. Serielles Zwischenblock-interface		
3.1. ZI-SE 3601	Zwischenblockinterface, Steuereinheit	Nahinterface
1) 3.2. ZI-UE 3602	Zwischenblockinterface, Uebertragungseinheit	Ferninterface
3.3. UEW 2338	Zentrale Ueberwachungskarte fuer Grundeinheit ursadat 5000	TTL-Pegel, 3 Eingaenge

1) Kein direkter Anschluss an K 1520 Systembus

Anlage 6: Anwendungsmoeglichkeiten der Prozess-Geraetetreiber

In den Tabellen dieser Anlage werden die Anwendungsmoeglichkeiten der Treiber, Interrupt-Service-Routinen und der Standard-Initialisierungs-Routinen fuer die Bedienung der Prozess-Steckkarten des Systems ursadat 5000 sowie die Typen der benutzten Zuweisungs- und E/A-Tabellen angegeben, deren Aufbau in der Anlage 8 (Zuweisungs- und E/A-Tabellen der Prozess-Geraetetreiber) dargestellt ist.
Die Tabellen der Anlage besitzen folgenden Aufbau:

Spalte 1: Treibertyp
Spalte 2: Treibername
Spalte 3: Name der zugeordneten Interrupt-Service-Routine
Spalte 4: Name der zugeordneten Standard-Initialisierungs-Routine
Spalte 5: Typ der Zuweisungs-Tabelle
Spalte 6: Typ der E/A-Tabelle
Spalte 7: Anwendungsbeschraenkungen

Dabei gilt die Systemfestlegung, dass die Sub-bzw. Kanaladressen jeweils in dem niederwertigen Byte der physischen Portadresse verschluesselt sind.

Anlage 6.1.: Ausgabe digital statisch

Typ	TREIBER ROUTINE	INTERRUPT-SERVICE-ROUTINE	INIT-ROUTINE	Typ Zuw. TAB.	TYP E/A-TAB.	Zulässige Subadressen der Prozeßsteckkarten					
						DAS-M (233Ø)	DA-R (2331)	DAS-KT (2334)	DA-O (2335)	DA-T (2336)	DEAS (2337)
DAS	EW.DASE	—	EW.IDASØ	1	1	—	Ø,1,2	—	—	Ø,1,2,3	—
			EW.IDASE	1	1	—	—	—	Ø1	—	—
			EW.IDEA	1	1•	—	—	—	—	—	Ø,1,4
			EW.INULL	1	1	Ø [1]	—	Ø 1,2,3 [1]	—	—	—
	EW.DASH	EW.DASHC	EW.IDASH	1	1Ø	Ø	—	—	—	—	—
	EW.DASKT	—	EW.INULL	1	1	—	—	Ø	—	—	—
	EW.DASD1	—	EW.IDASØ	1	1	—	Ø1	—	—	Ø,1,2	—
			EW.IDSD1	1	1	—	—	—	Ø	—	—
			EW.IDEA	1	1	—	—	—	—	—	Ø
			EW.INULL	1	1	—	—	Ø 1,2 [1]	—	—	—
	EW.DASD2	—	EW.IDASØ	1	1	—	Ø	—	—	Ø,1	—
	EW.DASB1	—	EW.IDASB	2	1	Ø [1]	Ø,1,2	Ø [1]	Ø,1	Ø,1,2,3	—
			EW.IDEA	1	1	—	—	—	—	—	Ø,1,4

Anlage 6.2. : Ausgabe digital dynamisch

Typ	TREIBER-ROUTINE	INTERRUPT-SERVICE - ROUTINE	INIT-ROUTINE	TYP Zuw.-TAB.	TYP E/A-TAB.	Zulässige Subadressen der Prozeßsteckkarten:		
						DA-R (2331)	DA-O (2335)	DA-T (2336)
DAD	EW.DADE	EW.DADEI	EW.IDADE	1	2	Ø,1	Ø,1	Ø,1
	EW.DADD1	EW.DADI 1	EW.IDDD1	1	2	Ø	Ø	Ø
	EW.DADD2	EW.DADI 2	EW.IDDD 2	1	2	Ø	–	Ø,1
	EW.DADB 1	EW.DABI 1	EW.IDADB	2	3	Ø, (2) [1] 1	Ø,1	Ø,(2) [1] 1,(3)

1) Soll eine Impulsausgabe über die im Klammern stehenden Subadressen erfolgen, so muß gleichzeitig eine Ausgabe über die vorangestellte Subadresse erfolgen

Anlage 6.3. : Bedienung Prozeßkarte DEAS

TYP	TREIBER - ROUTINE	INTERRUPT - SERVICE - ROUTINE	INIT - ROUTINE	TYP ZUW.-TAB.	TYP E/A - TAB	DEAS 2337
DEAS	EW. DACE	EW. DACEI	EW. IDEA	1	6	Ø,1
	EW. DACB	EW. DACBI	EW. IDEA	1	2 [1]	Ø,1
		EW. DBCBI	EW. IDBI	1	2 [1]	Ø
	EW. DECE	EW. DEUEI	EW. IDEA	1	16	Ø,1 4 [2]
	EW. DECB	EW. DECBI	EW. IDEA	1	7	Ø,1
		EW. DBCBI	EW. IDBI	1	7	Ø

1) Byte 4 ungenutzt

2) nur in Bit-Mode mit INT-Auslösung

Anlage 6.4.: Eingabe digital statisch

TYP	TREIBER-ROUTINE	INTERRUPT-SERVICE-ROUTINE	INIT-ROUTINE	TYP ZUW.-TAB.	TYP E/A-TAB.	Zulässige Subadressen der Prozeßsteckarten:				
						DEAS (2337)	DES (234Ø)	DEM (2341)	DES-KT (2344)	DAS-H (233Ø)
DES	EW.DESE	–	EW.IDESE	1	1	–	Ø1	–	–	–
			EW.IDEA	1	1	Ø1,4	–	–	–	–
			EW.INULL	1	1	–	–	–	Ø,1,2,3	Ø
	EW.DESD	–	EW.IDESD	1	1	–	Ø	–	–	–
			EW.IDEA	1	1	Ø	–	–	–	–
			EW.INULL	1	1	–	–	–	Ø,1,2	–
	EW.DESB1	–	EW.IDESB	2	1	–	Ø1	–	Ø,1,2,3	-Ø
			EW.IDE A	2	1	Ø,1,4	–	–	–	–
DEM	EW.DESME	EP.HIA 1)	EW.IDEM	1	15	–	–	Ø... 15 2)	–	–
	EW.DESMB			2						

1) <HL> - Adresse der E/A-Tabelle +1

2) Kanal - Adressen

Anlage 6.5.: Eingabe digital mit INT - Auslösung

TYP	TREIBER - ROUTINE	INTERUPT - SERVICE - ROUTINE	INIT - ROUTINE	TYP ZUW. - TAB.	TYP E/A - TAB.	zul. Subadressen der Prozeßkarten:	
						DES (234Ø)	DED (2342)
DESU	EW. DEUE	EW. DEUEI	EW. IDEUS EW. IDEUE	1	16	Ø1	-
	EW. DEUD	EW. DEUDI	EW.IDEUD	1	16	Ø	-
	EW. DEUB 1	EW. DUBI 1	EW. IDEUB	2	5	Ø1	-
DED	EW. DEDE	EW. DEDEI	EW. IDEDE	1	16	-	Ø1

Anlage 6.6.: Impuls-E/Ausgabe

Typ	TREIBER-ROUTINE	INTERRUPT-SERVICE-ROUTINE	INIT-ROUTINE	Typ ZUW.-TAB.	Typ E/A-TAB	Zul. Subadressen der IA (2339)	Zul. Subadressen der UIZ (2343)
	EW.IAZTE	EW.IAINI	EW.IAINI	1	4	Ø,1,2,3 [1]	–
	EW.IAZLE	EW.IAINI / EW.IAAKI	EW.IAINI	1	11	1,3 [1]	–
UIZ	EW.UIZ 4	EW.UIZI	EW.IUIZ	1	13	–	4,5,6,7

1) Kanal-Nr.

Anlage 6.7. : Analog Ein/Ausgabe

Typ	TREIBER-ROUTINE	INTERRUPT-SERVICE ROUTINE	INIT-ROUTINE	Typ ZUW.-TAB.	Typ E/A-TAB.	Zul. Kanaladressen für die Prozeßsteckkarten		
						AE-G (231Ø)	AA-5K (23Ø3)	AA-1K (23Ø1)
AEG	EW.AEG1	EW.AEGI.1	EW.IAEG1	3	8	Ø...55 [1]	–	–
	EW.AEG2	–	EW.IAEG2	3	9	Ø...55 [1]	–	–
AA5K	EW.AA5KU	–	EW.IAA51	1	1	–	1,2,4,5,6	–
	EW.AA5KB							
AA1K	EW.AA1K	EW.IAA1K	EW.IAA1K	1	12	–	–	Ø

[1] entsprechend der Ausstattung

Anlage 7: Rufbeschreibung der Prozess-Rufe

1. Rufe von Typ DAS

1.1. Notation

```
[name] POUT  R=n,(DAS=i[,ECA=(adr!symb)]
                       [,BOB=(adr!symb)]
                       [,(MASK=m!POM=(adr!symb))]
                       [,WAIT]     1)
                    !PARA=(adr!symb))[;Kommentar]

                             1) nur fuer DASH
```

1.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte),Modi-fikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Daten-Puffers	2	nur bei Angabe von BOB
Masken-Byte	1	nur bei Angabe von MASK
Zeiger auf Maske	2	nur bei Angabe von POM

1.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit:	7	6	5	4	3	2	1	0
	WAIT	x	x	x	POM	MASK POM	BOB	ECA

2. Rufe vom Typ DAD

2.1. Notation

```
[name] POUT  R=n,(DAD=i[,ECA=(adr!symb)]
                       [,BOB=(adr!symb)]
                       [,(MASK=m!POM=(adr!symb))]
                       [,REC=a]
                       [,MASTER=s]
                       [,BOT=(adr!symb)]
                       [,WAIT]
                      !PARA=(adr!symb))[;Kommentar]
```

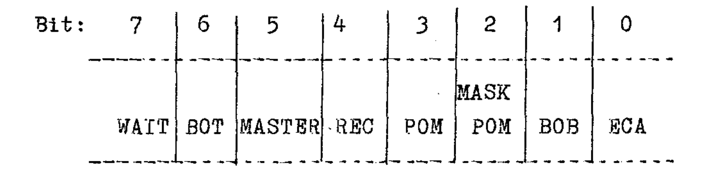
2.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameterblock	2	nur bei Angabe von PARA
		--- Rufparameterblock ---

Ruflaenge (-3 Byte), Modifikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Daten-Puffers	2	nur bei Angabe von BOB
Maskenbyte	1	nur bei Angabe von MASK
Adresse Maske	2	nur bei Angabe von POM
Anzahl der Bloecke	1	nur bei Angabe von REC
Port-Nr. des Masters	1	nur bei Angabe von MASTER
Adresse der Liste "Impulszeiten"	2	nur bei Angabe von BOT

2.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist.



Bit:	7	6	5	4	3	2	1	0
	WAIT	BOT	MASTER	REC	POM	MASK POM	BOB	ECA

3. Rufe vom Typ DES

3.1. Notation

```
[name] PIN  R=n,(DES=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,(MASK=m!POM=(adr!symb))]
                      [,(TAKT=t!POT=(adr!symb))]
                  !PARA=(adr!symb))[;Kommentar]
```

3.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte), Modifikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Maskenbyte	1	nur bei Angabe von MASK
Adresse Maske	2	nur bei Angabe von POM
Zeitkonstante Takt	1	nur bei Angabe von TAKT (1)
Adresse der Liste "Takt-zeiten	2	nur bei Angabe von POT (2)

(1) nur gueltig fuer DESE, DESB
(2) nur gueltig fuer DESBI

3.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit:	7	6	5	4	3	2	1	0
	0	x	x	TAKT POT	POM	MASK POM	BOB	ECA

4. Rufe vom Typ DESM

4.1. Notation

```
[name] PIN   R=n,(DESM=i[,ECA=(adr!symb)]
                        [,BOB=(adr!symb)]
                        [,(MASK=m!POM=(adr!symb))]
                        [,WAIT]
                  !PARA=(adr!symb))[;Kommentar]
```

4.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte), Modifikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Eingabe-Maske	2	nur bei Angabe von MASK
Zeiger auf Eingabe-Maske	2	nur bei Angabe von POM

4.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit:	7	6	5	4	3	2	1	0
	WAIT	x	x	x	POM	MASK POM	BOB	ECA

5. Rufe vom Typ DESU

5.1. Notation

```
[name] PIN   R=n,(DESU=i[,ECA=(adr!symb)]
                        [,BOB=(adr!symb)]
                        [,(MASK=m!POM=(adr!symb))]
                        [,(TAKT=t!POT=(adr!symb))]
                        [,USER=(adr!symb)]
                        [,BOT=(adr!symb)]
                   !PARA=(adr!symb))[;Kommentar]
```

5.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte), Modifikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Maskenbyte	1	nur bei Angabe von MASK

Adresse Maske	2	nur bei Angabe von POM
Zeitkonstante "Takt"	1	nur bei Angabe von TAKT (1)
Adresse der Liste "Taktzeiten"	2	nur bei Angabe von POT (2)
Adresse einer Anwender-ISR	2	nur bei Angabe von USER
Adresse der Liste "Verzoegerung"	2	nur bei Angabe von BOT

(1) nur zulaessig fuer die Treiber DEUE, DEUD
(2) nur zulaessig fuer die Treiber DEUBi

5.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist.

Bit: 7	6	5	4	3	2	1	0
0	BOT	USER	POT TAKT	POM	MASK POM	BOB	ECA

6. Rufe vom Typ DED

6.1. Notation

```
[name] PIN  R=n,(DED=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,USER=(adr!symb)]
                      [,MEMORY]
                  !PARA=(adr!symb))[;Kommentar]
```

6.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte), Modi-fikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Adresse einer Anwender-ISR	2	nur bei Angabe von USER

6.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleich-gueltig ist.

Bit:	7	6	5	4	3	2	1	0
	0	MEMORY	USER	x	x	x	BOB	ECA

7. Rufe fuer die Prozess-Steckkarte DEAS

7.1. Notation der Ausgabe

```
[name] POUT  R=n,(DAC=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,SIZE=1]
                      [,WAIT]
                  !PARA=(adr!symb))[;Kommentar]
```

7.1.1. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte), Modi-fikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Blocklaenge	1	nur bei Angabe von SIZE

7.1.2. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition = "1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleich-gueltig ist.

Bit:	7	6	5	4	3	2	1	0
	WAIT	x	x	SIZE	x	x	BOB	ECA

7.2. Notation der Eingabe

```
[name] PIN   R=n,(DEC=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,USER=(adr!symb)]
                  !PARA=(adr!symb))[;Kommentar]
```

7.2.1. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ----
Ruflaenge (-3 Byte), Modi-fikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Adresse einer Anwender-ISR	2	nur bei Angabe von USER

7.2.2. Belegung des Kommando-Bytes

Bit:	7	6	5	4	3	2	1	0
	x	x	USER	0	x	x	BOB	ECA

8. Ruf fuer die Bedienung der Prozess-Steckkarte IA

8.1. Notation

```
[name] POUT  R=n,(IA=i[,ECA=(adr!symb)]
                     [,BOB=(adr!symb)]
                     [,USER=(adr!symb)]
                     [,SERO=(adr!symb)]
                     [,WAIT]
                     [,BREAK]
                    !PARA=(adr!symb))[;Kommentar]
```

8.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameterblock	2	nur bei Angabe von PARA
		------Rufparameterblock--------
Ruflaenge (-3 Byte)	1	Bit=7; BREAK
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB
Adresse einer Anwender-ISR	2	nur bei Angabe von USER
Adresse einer Statusfehler-Routine	2	nur bei Angabe von SERO

8.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiviert, wenn die entsprechende Bitposition "1" ist. Kennzeichnung einer Bitposition mit "x" bedeutet nicht relevant.

Bit	7	6	5	4	3	2	1	0
	WAIT	SERO	USER	x	x	x	BOB	ECA

9. Ruf fuer die Bedienung der Prozess-Steckkarte UIZ

9.1. Notation

```
[name] PIN  R=n,(UIZ=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,(MASK=m!POM=(adr!symb))]
                      [,POT=(adr!symb)]
                      [,USER=(adr!symb)]
                    !PARA=(adr!symb))[;Kommentar]
```

9.2. Struktur

Bedeutung	Byteanzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameterblock	2	nur bei Angabe von PARA
		---- Rufparameterblock --------
Ruflaenge (-3 Byte), Modifikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes

Adresse des Fehler-bytes	2	nur bei Angabe von ECA
Adresse des Eingabe-puffers	2	nur bei Angabe von BOB
Maskenbyte	1	nur bei Angabe von MASK
Adresse Maske	2	nur bei Angabe von POM
Adresse der Liste "Zeitkonstanten"	2	nur bei Angabe von POT
Adresse der Anwender-ISR	2	nur bei Angabe von USER

9.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition gleich "1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit	7	6	5	4	3	2	1	0
	x	x	USER	POT	POM	MASK POM	BOB	ECA

10. Ruf fuer die Bedienung der Prozess-Steckkarte AE-G

10.1. Notation

```
[name] PIN  R=n,(AEG=i[,ECA=(adr!symb)]
                      [,BOB=(adr!symb)]
                      [,((MASK=m!POM=(adr!symb))
                           !MUX=(adr!symb))]
                      [,SCV]
                      [,WAIT]
                  !PARA=(adr!symb))[;Kommentar]
```

10.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameterblock	2	nur bei Angabe von PARA
		---- Rufparameterblock ------
Ruflaenge (-3 Byte), Modifikation	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Daten-Puffers	2	nur bei Angabe von BOB
Eingabemaske	1	nur bei Angabe von MASK
Zeiger auf Eingabemaske	2	nur bei Angabe von POM
Adresse der Liste "MUX-Adressen"	2	nur bei Angabe von MUX 1)

1) Aufbau der Liste "MUX-Adressen":

Byte-Nr.	Bedeutung
1	n=Anzahl der Multiplexer-Adressen
2	1. Multiplexer-Adresse
.	
.	
n+1	n. Multiplexer-Adresse

10.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition ="1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit	7	6	5	4	3	2	1	0
	x b.Polling WAIT	SCV	MUX	x	POM	MASK POM	BOB	ECA

11. Ruf fuer die Bedienung der Prozess-Steckkarte AA-1K

11.1. Notation

```
[name] POUT  R=n,(AA1K=i[,ECA=(adr!symb)]
                        [,BOB=(adr!symb)]
                        [,USER=(adr!symb)]
                    !PARA=(adr!symb))[;Kommentar]
```

11.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		------ Rufparameterblock ------
Ruflaenge (-3 Byte)	1	
Modifikation		
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung der Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe BOB
Adresse der Anwender-ISR	2	nur bei Angabe USER

11.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition ="1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit	7	6	5	4	3	2	1	0
	0	x	USER	x	x	x	BOB	ECA

12. Ruf fuer Bedienung der Prozess-Steckkarte AA-5K

12.1. Notation

```
[name] POUT  R=n,(AA5K=i[,ECA=(adr!symb)]
                         [,BOB=(adr!symb)]
                     !PARA=(adr!symb))[;Kommando]
```

12.2. Struktur

Bedeutung	Byte-anzahl	Bemerkungen
RST	1	
Ruf-Nr.	1	
Zeiger auf Rufparameter-block	2	nur bei Angabe von PARA
		---- Rufparameterblock ------
Ruflaenge (-3 Byte)	1	
logische Geraete-Nr.	1	
Kommando-Byte	1	siehe Belegung des Kommando-Bytes
Adresse des Fehlerbytes	2	nur bei Angabe von ECA
Adresse des Datenpuffers	2	nur bei Angabe von BOB

12.3. Belegung des Kommando-Bytes

Die angegebenen Funktionen des Kommando-Bytes sind dann aktiv, wenn die entsprechende Bitposition ="1" ist. Die Kennzeichnung einer Bitposition mit "x" bedeutet, dass die Belegung gleichgueltig ist.

Bit	7	6	5	4	3	2	1	0
	0	x	x	x	x	x	BOB	ECA

Anlage 8: Zuweisungs- und E/A-Tabellen der Prozess-Geraete-treiber

1.1. Zuweisungstabellen

Typ 1

Byte	Inhalt	Bemerkungen
0,1	Adresse der E/A- Tabelle	
2,3	Adresse der Initialisierungsroutine	
4,5	Adresse des Treibers	
6	physische Portadresse	
7,8	Standard- Pufferadresse	

Typ 2

Byte	Inhalt	Bemerkungen
0,1	Adresse der E/A- Tabelle	
2,3	Adresse der Initialisierungsroutine	
4,5	Adresse des Treibers	
6,7	Adresse des Portverzeichnisses	Aufbau s.u.
8,9	Standard- Pufferadresse	
Aufbau des Portverzeichnisses		
0	Anzahl der Ports des log. Geraetes	
1	physische Adresse Port 1	Port- Nr. 1
2	physische Adresse Port 2	Port- Nr. 2
...		
n	physische Adresse Port n	Port- Nr. n

Typ 3

Byte	Inhalt	Bemerkungen
0,1	Adresse der E/A- Tabelle	
2,3	Adresse der Initialisierungsroutine	
4,5	Adresse des Treibers	
6	physische Adresse der Prozess- Karte	AB0 - AB2 = "0"
7	hoechste Multiplexer- Adresse	max. 55
8,9	Standard- Pufferadresse	

Typ 4

Byte	Inhalt	Bemerkungen
0,1	Adresse der E/A- Tabelle	
2,3	Adresse der Initialisierungsroutine	
4,5	Adresse des Treibers	
6	physische Adresse des Moduls	
7,8	Adresse der variablen Initialisierungsparameter	

1.2. E/A- Tabellen

Typ 1

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7 = 0 (ohne Wait)
1	Status- Byte	Bit 7 = 0 (Polling-Betrieb) Bit 3 = 1 (anlaufinitialisiert)

Typ 2

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7 = 1 (Ruf mit Wait)
1	Status- Byte	Bit 7 = 1 (Warte- Betrieb) Bit 3 = 1 (anlaufinitialisiert) Bit 0 = 1 (Geraet besetzt)
2	rufende Task	
3	wartende Task bzw. 0	
4	physische Adresse des Interrupt- Senders	E/A- Arbeitstabelle
5	Blockzaehler	
6	physische (1.) Port- adresse	
7,8	aktuelle Pufferadresse	

TYP 3

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT
1	Status- Byte	Bit 7=1 (Warte- Betrieb) Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt
2	rufende Task	
3	wartende Task bzw. 0	

4	phys. Adresse des Interrupt- Senders	E/A- Arbeitstabelle
5	Blockzaehler	
6,7	Adresse Portverzeichnis	
8	Ausgabemaske	
9,10	Aktuelle Pufferadresse	

TYP 4

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT Bit 5=1 : Anwender- ISR
1	Status- Byte	Bit 7=1 (Warte- Betrieb) Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt
2	rufende Task	
3	wartende Task bzw. 0	
4,5	Adresse der anwender-spezifische ISR	E/A- Arbeitstabelle
6	phys. (1.) Port-Adresse	
7,8	aktuelle Pufferadresse	

TYP 5

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=0 (Polling- Betrieb) Bit 5=1 : Anwender- ISR im Ruf

1	Status- Byte	Bit 7=0 Bit 3=1 :anlaufinitialisiert
2,3	Adresse einer anwenderspezifischen ISR	E/A- Arbeitstabelle
4,5	Adresse Portverzeichnis	
6	aktuelle Eingabemaske	
7,8	aktuelle Pufferadresse	

TYP 6

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT
1	Status- Byte	Bit 7=1 (Warte- Betrieb) Bit 3=1 : anlaufinitialisiert Bit 0=0 : Geraet besetzt
2	rufende Task	
3	wartende Task bzw. 0	

TYP 7

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=0 (ohne WAIT)
1	Status- Byte	Bit 7=1 : Warte- Betrieb Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt

2	rufende Task	
3	wartende Task bzw. 0	
4,5	Adresse einer anwenderspezifischen ISR	E/A- Arbeitstabelle
6	physische Portadresse	
7	Blocklaenge	
8,9	aktueller Pufferzeiger	
10,11	Pufferadresse	

TYP 8

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT
1	Status- Byte	Bit 7=1 (Warte- Betrieb) Bit 5=1 : Geraete- Ueberwachung Bit 4=1 : Ueberwachungszyklus Bit 3=1 : anlaufinitialisiert Bit 2=1 : Adress- Fehler Bit 2=0 : Geraete- Fehler Bit 1=1 : Fehler Bit 0=1 : Geraet besetzt
2	rufende Task	L
3	wartende Task bzw. 0	
4,5	Adresse Fehler- Byte	E/A- Arbeits- Tabelle
6	Restanzahl der Multiplexer- Adressen	
7	akt. MUX- Adresse zur akt. Tabellenadresse	

8	akt. Maskenbyte	
9	physische Adresse der Prozesskarte	
10,11	aktuelle Pufferadresse	
12,13	Pruefwert	Festkommazahl
14	zul.Fehler in Promille	
15,16	aktuelle Maskenadresse	
17	aktuelle logische Geraete- Nummer	

TYP 9

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=0 (ohne WAIT)
1	Status- Byte	Bit 7=1 Bit 5=1 : Kontrollzyklus Bit 3=1 : anlaufinitialisiert Bit 2=1 : Adressfehler Bit 2=0 : Geraetefehler Bit 1=1 : Fehler Bit 0=1 : Geraet besetzt
2,3	Adresse Fehlerbyte	E/A- Tabelle
4	Restanzahl der Multiplexer- Adressen	
5,6	Pruefwert	Festkommazahl
7	zulaessige Abweichung in Promille	
8	aktuelle logische Geraete- Nummer	
9	hoechste MUX- Adr. akt. Eingabemaske	

TYP 10

Byte	Inhalt	Bemerkungen
1	Ueberwachungs- Nr.	fuer Zeitsteuerung
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT
1	Status- Byte	Bit 7=1 : Wartebetrieb Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt
2	rufende Task	
3	wartende Task	
4,5 6 7 8	Adresse Fehlerbyte physische Portadresse Ausgabewert logische Geraete- Nr.	E/A- Arbeitstabelle

TYP 11

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT Bit 5=1 : Anwender- ISR
1	Status- Byte	Bit 7=1 : Warte- Betrieb Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt
2	rufende Task	
3	wartende Task bzw. 0	

4,5	Adresse einer anwenderspezifischen ISR	E/A- Arbeitstabelle
6	physische Kanaladresse	
7,8	aktuelle Pufferadresse	
9	aktuelles initialisiertes Steuerwort des Takt- CTC	des dem Kanal zugeordneten Taktsystems, wenn Impulszaehlausgabe aktiviert
11	Zaehler Anlaufaenderungspunkte	

TYP 12

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=0 : (ohne WAIT) Bit 5=1 : Anwender ISR
1	Status- Byte	Bit 7=1 Bit 6=1 x2= ext. Schalter "EIN" Bit 5=1 x1= 15V ausgefallen Bit 4=1 x0= AA1K ueber Programm ext. geschaltet Bit 3=1 : anlaufinitialisiert
2,3	Adresse Fehlerbyte	E/A- Arbeitstabelle
4,5	Adresse einer anwenderspezifischen ISR	
6	phys. Adresse des Interrupt- Senders	Port A des AA - 1K

TYP 13

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT Bit 5=1 : Anwender ISR

1	Status- Byte	Bit 7=1 : Warte- Betrieb Bit 3=1 : anlaufinitialisiert Bit 2=1 : Kettung Kan. 2-3 Bit 1=1 : Kettung Kan. 0-1 / 0-1-2 / 0-1-2-3 Bit 0=1 : Geraet besetzt
2	rufende Task	
3	wartende Task	
4,5	Adresse einer anwenderspezifischen ISR	E/A- Arbeitstabelle
6	phys. Adresse Kanal 0	
7	aktuelle Eingabemaske	
11	Datenwort PIO- Port A	
12	Datenwort PIO- Port B	
13	Steuerwort	
14	Zeitkonstante Kanal 0	
15	Steuerwort	
16	Zeitkonstante Kanal 1	
17	Steuerwort	
18	Zeitkonstante Kanal 2	
19	Steuerwort	
20	Zeitkonstante Kanal 3	
21	Ueberlaufzaehler Kanal 0	
22	Ueberlaufzaehler Kanal 1	
23	Ueberlaufzaehler Kanal 2	
24	Ueberlaufzaehler Kanal 3	

TYP 14

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT Bit 5=1 : Anwender- ISR
1	Status- Byte	Bit 7=1 : Warte- Betrieb Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt
2	rufende Task	
3	wartende Task bzw. 0	
4,5 6 7,8	Adresse einer anwen- derspezifischen ISR pys. Adresse des Moduls Adresse der variabelen Initialisierungs- parameter	E/A- Arbeitstabelle

TYP 15

Byte	Inhalt	Bemerkungen
1	Ueberwachungs- Nummer	fuer Zeitsteuerung
0	Kommando- Byte	Bit 7=1 : Ruf mit WAIT
1	Status- Byte	Bit 7=1 : Warte- Betrieb Bit 3=1 : anlaufinitialisiert Bit 0=1 : Geraet besetzt
2	rufende Task	

3	wartende Task bzw. 0	

TYP 16

Byte	Inhalt	Bemerkungen
0	Kommando- Byte	Bit 5=1 : Anwender- ISR
1	Status- Byte	Bit 7=0 Bit 0=0
2,3	Adresse einer anwenderspezifischen ISR	E/A- Arbeitstabelle
4	phys. (1.) Portadresse	
5,6	akt. Pufferadesse	

Abkuerzungsverzeichnis

Abkuerzung		Bedeutung
ABS	–	Anschlusssteuerung Bildschirm
AFS	–	Anschlusssteuerung Folienspeicher
AP	–	Applikationsprogramm
BCES	–	Anfangssektordatei (begin of extent sector)
BOET	–	Anfangsspurdatei (begin of extent track)
CL	–	Grundtakt (clock)
CPU	–	Zentrale Recheneinheit (central prozessing unit)
CRC	–	Pruefrechnung
CTC	–	Zeitschaltkreis
DEL	–	Delete-Taste
DSB	–	Dateisteuerblock
DV	–	Datenverarbeitung
DVNB	–	Geraetenummer
E/A	–	Ein-/Ausgabe
EIEX	–	internspeicherorientiertes Echtzeitsteuerprogrammsystem
FD	–	Floppy-Disk
HDR	–	Dateikennsatz
HR	–	Stunde
INS	–	Insert-Taste
I/O	–	Input/Output
ISR	–	Interrupt-Service-Routine
KES	–	Karteneinschub
KROS	–	Kombinat Robotron-Standard
MF	–	Folienspeicherlaufwerk MOM-Flex
MI	–	Minute
MON1	–	Bildschirmbaugruppe 1
MOS	–	Maschinenorientierte Systemunterlagen
MRS	–	Mikrorechnersystem
MS	–	Millisekunde

OEM	–	Original equipment manufacturer (Original Geraetehersteller)
PIO	–	Parallel input/output (Parallel-Interface-Schaltkreis)
PROM	–	programmierbarer Nur-Lese-Speicher
RAM	–	Lese- und Schreibspeicher (random access memory)
SE	–	Sekunde
SD	–	Seriendrucker
SIF	–	Standardinterface
SYPS		
K 1520	–	Assemblersprache K 1520
TTL	–	Transistor-Transistor-Logik-Schaltkreis (transistor-transistor-logic)
UP	–	Unterprogramm
VOL	–	Dateitraegerkennsatz
ZRE	–	Zentrale Recheneinheit

Sachwortverzeichnis

Druck und buchbinderische Weiterverarbeitung: TASTOMAT, Eggersdorf
Ag 706/303/83 – 8152